DISS. ETH N° 24052

Near-optimal Adaptive Information Acquisition: Theory and Applications

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

YUXIN CHEN

M. Sc. in Computer Science, The University of Kansas

born on 29.05.1991

citizen of China

accepted on the recommendation of

Prof. Dr. Andreas Krause (ETH Zurich), examiner

Prof. Dr. Joachim Buhmann (ETH Zurich), co-examiner

Prof. Dr. Kilian Weinberger (Cornell University), co-examiner

2017

Abstract

Many problems in artificial intelligence require adaptively making a sequence of decisions, each based on the information obtained so far. As an example, consider interactive troubleshooting, where one can perform many tests on the system in question, and the goal is to adaptively conduct a small set of tests that are sufficient for making the correct diagnosis. Another example is active perception in robotics, where the goal is to localize a certain object by exploring the environment via some sensing actions, such as touch or vision. Understanding how to effectively acquire information under partial observability is fundamental for developing intelligent adaptive systems. Unfortunately, most of the existing techniques for solving such problems are either simple heuristics which do not strive for optimality or principled non-myopic approaches that are difficult to scale to larger problems.

To ease the tension between theory and practice, this dissertation pursues the fundamentals of adaptive information acquisition, with the goal to devise a mathematical and algorithmic framework for efficient (in terms of computational complexity) and robust (in terms of decision quality and noise-tolerance) decision making under uncertainty. From the theoretical perspective, we look into different problem settings, where it is challenging to characterize the *value* of information due to complex constraints and modeling assumptions, such as *indirect information*, *uncertain inputs*, *delayed feedback* in parallel systems, and *incomplete knowledge* about the model. We provide new theoretical insights and develop novel algorithms for solving such problems. From the practical perspective, we demonstrate strong empirical performance for our proposed algorithms on a number of problem instances, including Bayesian experimental design for behavioral economics, interactive troubleshooting, active preference learning, active touch-based localization, and active object detection for biodiversity monitoring.

More specifically, in Part II of this dissertation, we investigate *sequential algorithms*

which aim to optimize the *value of information*. We first look into a basic variant of the adaptive information acquisition problem, where the goal is to learn the value of some target random variable through a sequence of conditionally independent, possibly noisy tests. Here, the value of information is defined in terms of the informativeness of the tests performed, measured by Shannon's mutual information. We provide the first rigorous analysis of the greedy algorithm that holds under the persistent noise setting.

In most practical applications, collecting information is not the goal of its own, but rather a means for making informed decisions. We further investigate novel, efficient objectives for such problems which are amenable to greedy optimization. In particular, we propose novel surrogate objectives that are: (1) aligned with the value of information problem (2) efficient to evaluate and (3) *adaptive submodular*. This latter property enables us to utilize an efficient greedy optimization while providing strong approximation guarantees. Our algorithms achieve the state-of-the-art performance on a few problem instances including a real-world robotic manipulation task.

Moving beyond the previous settings, we seek to generalize our theoretical insight of constructing submodular surrogates for solving more general sequential decision problems. We propose a principled approach to active object detection and show that for a rich class of base detection algorithms, one can derive a natural sequential decision problem for deciding when to invoke expert supervision. We demonstrate the effectiveness of our algorithm on several object detection problems.

Part III of this dissertation aims to address some practical challenges of the adaptive information acquisition problem. For instance, in many practical scenarios, fully sequential selection could be infeasible. We study information-parallel learning and decision making. We prove that, for stochastic optimization problems which exhibit adaptive submodularity, a simple approach which greedily selects examples within a batch and assembles batches in a greedy manner, is competitive with the optimal batch-mode algorithm. Under certain assumptions, it is even competitive with the optimal sequential algorithm. Moreover, in practice, the underlying model that defines the objective function may be unavailable. We investigate the online sequential information acquisition problem, where parameters of the probabilistic model are initially unknown, and can only be learned from data in an online fashion. We establish a rigorous bound on the expected regret (defined in terms of the value of information) of our framework and demonstrate our algorithm on an online interactive troubleshooting application.

Zusammenfassung

Viele Probleme der künstlichen Intelligenz erfordern das Treffen einer Reihe von Entscheidungen, wobei jede Entscheidung auf den bislang gewonnenen Informationen basiert. Ein Beispiel hierfür ist die interaktive Fehlersuche, bei der eine Vielzahl an Tests in einem untersuchten System ausgewertet werden können; das Ziel ist jedoch, adaptiv eine geringe Anzahl an Tests auszuwählen, welche ausreichend sind um die richtige Diagnose zu stellen. Ein anderes Beispiel ist die gesteuerte Wahrnehmung in der Robotik, die das Ziel hat ein bestimmtes Objekt zu lokalisieren, indem die Umgebung mit visuellen oder haptischen Sensoren untersucht wird. Für die Entwicklung intelligenter, adaptive Systeme ist das Verständnis der effizienten Informationsgewinnung unter eingeschränkter Beobachtbarkeit von fundamentaler Bedeutung. Bedauerlicherweise sind die meisten existierenden Lösungsverfahren für diese Probleme entweder einfache Heuristiken, die keine optimale Lösung anstreben, oder basieren auf nicht-myopischen Herangehensweisen, welche sich schlecht auf Massendaten skalieren lassen.

Um die Differenzen zwischen Theorie und Praxis zu reduzieren, untersucht diese Dissertation die Grundlagen der adaptiven Informationsgewinnung, mit dem Ziel ein mathematisches und algorithmisches Framework für effiziente (in Sinne der Berechenbarkeit) und robuste (im Sinne der Qualität der getroffenen Entscheidungen und gegen statistisches Rauschen) Informationsgewinnung unter statistischer Unsicherheit zu entwerfen. Von einem theoretischen Standpunkt aus untersuchen wir verschiedene Problemstellungen, wobei aufgrund komplexer Randbedingungen und Annahmen an das Modell, wie indirekter Informationen, unsicherer Eingaben, verzögerter Rückmeldungen in parallelen Systemen und unvollständigem Wissen über das Modell, die Herausforderung darin besteht, den Wert der gewinnbaren Informationen zu beschreiben. Wir präsentieren neue theoretische Erkenntnisse und entwickeln neuartige Algorithmen fuer das Lösen der zuvor beschriebenen Probleme. Von einer praktis-

chen Perspektive aus demonstrieren wir herausragende empirische Ergebnisse unserer entwickelten Algorithmen auf unterschiedlichen Probleminstanzen. Die betrachteten Probleminstanzen umfasse unter anderem bayessche statistische Versuchsplanung in der Verhaltensökonomie, interaktive Fehlersuche, aktives Lernen von Vorlieben, aktive Lokalisierung mit Berührungssensoren, und aktive Objekterkennung für die Überwachung einer Artenvielfalt.

In Teil II dieser Dissertation untersuchen wir insbesondere sequentielle Algorithmen, die darauf abzielen den Wert der Informationen zu optimieren. Zuerst untersuchen wir eine einfache Variante des Problems der adaptiven Informationsgewinnung, mit dem Ziel basierend auf einer Sequenz von bedingt statistisch unabhängigen, möglicherweise verrauschten, statistischen Tests, den Wert einer bestimmten Zufallsvariable zu lernen. Hierbei wird der Informationswert der ausgeführten Tests mit Hilfe von Shannons Transinformation gemessen. Wir stellen die erste rigorose Analyse eines Greedy-Algorithmus vor, die auch unter persistentem statistischen Rauschen gilt.

In den meisten praktischen Anwendungen ist die Informationsgewinnung nicht das eigentliche Ziel, sondern ein Hilfsmittel um informierte Entscheidungen zu treffen. Daher untersuchen wir neue, effiziente Zielfunktionen für derartige Probleme, welche mittels Greedy-Algorithmen (approximativ) optimiert werden können. Insbesondere schlagen wir neue Ersatz-Zielfunktionen vor, welche (1) direkt mit dem Problem der Informationsgewinnung zusammenhängen (2) effektiv auszuwerten und (3) adaptiv submodular sind. Die letztgenannte Eigenschaft erlaubt es uns effektive Greedy-Optimierungsmethoden mit guten Approximationsgarantien anzuwenden. Unsere Algorithmen erzielen Ergebnisse auf dem neusten Stand der Technik auf verschiedenen Probleminstanzen, unter anderem bei der Steuerung von echten Robotern.

Über die vorangegangenen Problemstellungen hinaus ist es unser Ziel unsere theoretischen Einblicke in die Konstruktion von submodularen Ersatz-Zielfunktionen zu verallgemeinern und somit das Lösen allgemeinerer, sequentieller Entscheidungsprobleme zu ermöglichen. Wir schlagen eine prinzipielle Herangehensweise für aktive Objekterkennung vor, und leiten ein natürliches sequentielles Entscheidungsproblem für das Einbringen von Expertenwissen her. Wir demonstrieren die Effektivität unseres Algorithmus auf mehreren Objekterkennungsproblemen.

Teil III dieser Dissertation hat das Ziel einige praktische Herausforderungen des Problems adaptiven Informationsgewinnung zu lösen. Zum Beispiel ist in vielen praktischen Szenarien eine konsequent sequentielle Entscheidungsfindung nicht umsetzbar. Wir

untersuchen wie Informationen parallel gelernt und Entscheidungen parallel getroffen werden können. Wir zeigen das für stochastische Optimierungsprobleme mit adaptiver Submodularität eine einfache Greedy-Herangehensweise, welche nach dem Greedy-Prinzip Beispiele aus einem Stapelspeicher auswählt und diese nach dem gleichen Prinzip aufbaut, mit dem optimalen Stapelspeicher-Modus Algorithmus vergleichbar ist. Trifft man weitere Annahmen, ist er sogar mit der optimalen sequentiellen Strategie vergleichbar. Des Weiteren ist in der Praxis oft das der Zielfunktion zugrundeliegende Modell nicht verfügbar. Wir untersuchen das sequentielle Problem der Online-Informationsgewinnung mit unbekannten statistischen Parametern, die nur von den Daten während der Interaktion gelernt werden können. Für unser Framework beweisen wir eine Schranke für den Erwartungswert des Regrets (definiert mit Hilfe des Informationswerts) und demonstrieren die Anwendung unseres Algorithmus im Fall der interaktiven Fehlersuche.

Acknowledgments

This dissertation marks the end of a long and eventful journey, which would never have been possible without the support of many people. I want to express my sincere gratitude and appreciation to those who have contributed to this dissertation in various ways – either professionally or personally – along the way.

First and foremost, I owe my deepest gratitude to my Ph.D. supervisor Prof. Andreas Krause, for his unrelenting support in the past five years and continuous optimism concerning this work. Andreas has guided me through the research process with extreme patience and inspired me to "actively learn" from the frontier of the field. His clear vision, depth of thoughts, clarity of communication and enthusiasm about research not only helped me shape this dissertation, but also motivated me to grow as a full-fledged researcher. Andreas, thank you! I simply cannot imagine a better advisor.

I would like to extend my sincere thanks to my dissertation committee members, Prof. Joachim Buhmann and Prof. Kilian Weinberger, for carefully reading through my earlier drafts, and providing many constructive comments which greatly help me refine this dissertation. I would also like to thank my master thesis advisors, Prof. Bo Luo and Prof. Xue-wen Chen, for their constant encouragement to pursue this Ph.D. degree.

I am very grateful to the Computer Science department at ETH Zurich for providing me with an ideal platform for conducting my research. Special thanks go to Google for supporting me with the Google European Doctoral Fellowship for the last three years of my Ph.D. study. I am grateful to my mentors and colleagues at Microsoft Research Cambridge (Marc Brockschmidt, Daniel Tarlow, Pushmeet Kohli, Byron Cook), Xerox Research Center Europe (Morteza Chehreghani, Jean-Michel Renders), Google Zurich (Jurgen Van Gael), and MIT (John Fisher), for hosting my visits, as well as for the insightful discussion and career advices.

I gratefully acknowledge the contributions of all my coauthors – Andreas Krause, Amin Karbasi, Hamed Hassani, Gábor Bartók, Hioraki Shioi, César Fuentes Montesinos, Lian Pin Koh, Serge Wich, Shervin Javdani, Drew Bagnell, Marc Brockschmidt, Daniel Tarlow, Pushmeet Kohli, Byron Cook, Morteza Chehreghani and Jean-Michel Renders – It is very enjoyable to work with all of you. I am fortunate to work with and learn from many excellent master students (Nan, César, Hiroaki, Victor, Johannes and Sid). Thank you all for the work that eventually becomes part of this dissertation (and beyond).

I would like to thank all the members of the Learning and Adaptive Systems group, including Rita, Hasta, Adish, Gábor, Baharan, Mario, Amin, Alkis, Josip, Olivier, Hamed, Sebastian, Felix, Aytunç, Matteo, Johannes, Kfir, Besa and Andrew, for the countless hours in meetings, retreats, reading groups, dinners and group outings. It is through you that I felt myself connected to an active community in this foreign country.

A very special thank you to our group admin, Rita, for her untiring support, starting even before I came to Switzerland. Rita has always been a tremendous help no matter the task or circumstance, and she has routinely gone beyond her duty to help me out whenever I am frustrated with my poor knowledge of the (Swiss-) German language. Special thanks to Hasta and Adish who have worked by my side through the most time of my Ph.D., for the interesting technical discussions and moral support. Thanks to Hamed and Amin, for sharing thoughts and advice not only on research but also on long-term career development. Thanks to Johannes and Sebastian for their effort in proof-reading part of the manuscript and helping me with the German translation of the abstract. I am also grateful to the fun discussions with Cheng, Johann (for being such a patient German teacher), Brian, Nico and Dwarik.

I would like to say a heartfelt thank you to my family: to my parents, for always being the steadfast pillar of my support. For this and much more, I am forever in their debt. I am thankful to my brother for sharing his stories from another continent and being a source of inspiration. Thanks to my beloved grandmother and grandfather, who sadly left me a few years ago while I was in Switzerland. The images of my grandmother are filled in my childhood memories – It is to her I dedicate this dissertation. I would also like to thank the newest additions to my family, my wife and her family for their encouragement when writing this dissertation. Lastly, my utmost appreciation goes to my wife and dearest friend, Jiajia, for her caring, understanding and unconditional love during my good and bad times. Jiajia, thank you for sharing this ride with me! You have undoubtedly made the past five years the most precious memory of my life.

Contents

List of Figures

List of Tables

List of Acronyms

ActDet	Active Detection , 17, 119, 126, 132, 176
AL	Active Learning , 35
BatchGreedy	Batch-mode Greedy , 15, 17, 150, 152–156, 158, 177
BMAL	Batch Mode pool-based Bayesian Active Learning , 143, 155
CPT	Conditional Probability Table , 87, 167, 170
DAG	Directed Acyclic Graph , 89
DiRECt	Decision Region Edge Cutting , 11, 12, 15, 17–19, 68, 69, 82, 84, 86, 87, 91, 92, 94, 96, 103, 104, 106, 108, 109, 113, 114, 163, 176
DRD	Decision Region Determination , xix, 11, 19, 69, 70, 72–75, 78, 80, 82–87, 94, 95, 103–105, 108, 114, 117, 167
EC^2	Equivalence Class Edge Cutting , xix, 78–81, 83–87, 92, 93, 99–104, 106, 111, 114, 170, 202, 203, 207, 211, 221, 230–232
ECD	Equivalence Class Determination , xix, 78–80, 82–86, 94, 96, 98, 104, 106, 109, 114, 115
ECED	Equivalence Class Edge Discounting xiv, xix, xxii, 12, 17, 19, 69, 96, 97, 99–104, 106, 109, 110, 114, 115, 163, 177, 205–207, 229, 231, 233, 234, 236
GBS	Generalized Binary Search , 47, 50, 77, 78, 82, 103, 104, 106, 114, 143, 168
HEC	Hyperedge Cutting , 18, 19, 80–83, 94, 103, 106, 109, 113, 114, 176
IG	Information Gain , 104, 111, 169
MIS	Most Informative Selection , 9, 15, 17, 19, 48, 50, 65, 68, 74, 114
NVOI-NMU	Nonmyopic value of information problem for achieving near-maxmial utility , xxiv, 71–74, 82, 84, 87, 94, 114, 167
ODT	Optimal Decision Tree , 114

OnlineVoI	Online Optimal Value of Information , 16, 17, 162, 166
POMDP	Partially Observable Markov Decision Process , 167, 249
US	Uncertainty Sampling , 104, 106, 111, 169
VoI	Value of Information , 68, 70, 74, 88, 103, 104, 106, 111, 113

List of Notations

Universal notations

$\mathbb{H}(\cdot)$	Entropy of a distribution.
$\mathbb{H}_2(x)$	Binary entropy function: $\mathbb{H}_2(x) = -x \log x - (1-x) \log(1-x)$.
$\mathbb{H}(\cdot \mid \cdot)$	Conditional Entropy.
$\mathbb{1}\{\cdot\}$	Identity function.
$\mathbb{I}(\cdot;\cdot)$	Mutual information.
$\mathbb{P}[\cdot]$	Probability function.
$\mathbb{R}$	Real value.
$f(\mathbf{x}_\mathcal{A})$	Reward function of a set of observations $\mathbf{x}_\mathcal{A}$, objective function.
$F(\pi)$	Expected Reward of a policy π, objective function.
Q	Quota; The constraint on maximal reward to be achieved.
$\Delta(v \mid \mathbf{x}_\mathcal{A})$	Expected marginal benefit of test v, conditioned on observations $\mathbf{x}_\mathcal{A}$.
cost	Cost of a policy.
$\text{cost}_{avg}(\pi)$	The expected cost of a policy.
$\text{cost}_{wc}(\pi)$	The worst-case cost of a policy.
$c(v)$	Cost of test v.
$C(\mathcal{A}, \mathbf{x}_\mathcal{A})$	Cost of observing a set of test $\mathbf{x}_\mathcal{A}$.
$\mathcal{H}$	Set of hypotheses; version space.
$H, \mathbf{X}_\mathcal{V}$	Random variable that denotes a hypothesis; random variable that denotes the state of the world.
$h, \mathbf{x}_\mathcal{V}$	Hypothesis; a specific realization of the outcomes of all tests.
s	Size of the hypothesis set; number of hypotheses.
m	Size of the set of the root-causes.
n	Size of the domain of the target variable; number of decisions.
t	Size of the set of tests.

ψ	The partial realization, i.e., set of test-observation pairs observed by running policy π; $\mathbf{x}_{\mathcal{A}}$.
ψ_ℓ	Partial realization of the first ℓ tests selected by a policy; a path from the root to a node at level/depth ℓ of a decision tree.
Ψ	Random variable encoding a partial realization, i.e., set of test-observation pairs.
π	Policy, i.e., a (partial) mapping from observation vectors to tests.
$\pi_{[k]}$	Policy under budget k; policy of length k.
π^*	An optimal policy, either with the lowest cost for achieving some reward, or with the maximal reward under some budget.
OPT	π^*: an optimal policy.
$\mathcal{S}(\pi, h)$	The set of tests and outcomes seen by policy π, if hypothesis h is realized.
Π	Set of policies.
Θ	Random Variable that represents the root-cause.
θ	Value of root-cause.
$\text{supp}(\Theta)$	The ground set / domain of root-causes.
Y	Random variable that represent the target variable/decision.
y	Target variable/decision.
$\mathcal{Y}$	The domain of the target variable.
$\mathcal{V}$	The ground set of tests.
$\mathcal{A}, \mathcal{B}$	(sub-) sets of tests.
v	Test; candidate for object detection (§5).
X_v	Random variable that represents the outcome of test v.
x_v	A specific outcome of test v.
$\mathbf{x}_{\mathcal{A}}, \mathbf{x}_{\mathcal{B}}$	Observation of the outcomes of tests in $\mathcal{A}$, $\mathcal{B}$.

Notations specific to §3

D_v	Random variable that represent the deterministic outcome of test v.
d_v	Noise-free outcome of test v.
N_v	Noise on the outcome of test v.

Notations specific to §4

$w(E)$	Weight of edge $E \in \mathcal{E}$ in the EC^2/DIRECT graph.

$p_{\text{ERR}}^{\text{MAP}}(\psi)$	Error probability (of a MAP decoder), having observed partial realization ψ.
$p_{\text{ERR}}(\pi)$	$\mathbb{E}_{\psi_\pi}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\pi)\right]$, expected error probability by running policy π.
$\tilde{\mathcal{H}}$	Samples of hypotheses.
$\epsilon_{\theta,v}$	$1 - \arg\max_v \mathbb{P}\left[X_v = x_v\right]$: the noise rate for a test v.
f_{AUX}	The auxiliary function defined in Equation (4.5.3).
$\Delta_{\text{AUX}}(v \mid \psi)$	The expected gain in f_{AUX} by performing test v, conditioned on partial realization ψ.
$\Delta_{\text{ECED}}(v \mid \psi)$	The ECED gain which is myopically optimized at each iteration of the ECED algorithm.
$\Delta_{\text{ECED},\psi}(v)$	The re-normalized version of $\Delta_{\text{ECED}}(x_v \mid \psi)$, i.e., $\Delta_{\text{ECED},\psi}(v) = \Delta_{\text{ECED}}(v \mid \psi)/\mathbb{P}\left[\psi\right]^2$.
$\delta_{\text{OFFSET}}(x_v \mid \psi)$	The "offset" component in the ECED gain by observing x_v, having observed ψ.
f_{EC^2}	The EC^2 objective with $f_{\text{EC}^2}(\emptyset) := \sum_{\theta,\theta' \in \mathcal{E}} \mathbb{P}\left[\theta\right] \mathbb{P}\left[\theta'\right]$.
$f_{\text{EC}^2,\psi}$	The EC^2 objective with $f_{\text{EC}^2,\psi}(\emptyset) := \sum_{\theta,\theta' \in \mathcal{E}} \mathbb{P}\left[\theta \mid \psi\right] \mathbb{P}\left[\theta' \mid \psi\right]$.
$\Delta_{\text{EC}^2}(v \mid \psi)$	The EC^2 gain which is myopically optimized at each iteration of the EC^2 algorithm.
$\Delta_{\text{EC}^2,\psi}(v)$	The expected gain in $f_{\text{EC}^2,\psi}$ by performing test v, and *cutting* edges weight according to EC^2. It can be interpreted as $\Delta_{\text{ECED},\psi}(v)$, as if the test's outcome is noise-free, i.e., $\forall\theta,\ \epsilon_{\theta,e} = 0$.
$c_{\eta,\epsilon},\ c_\epsilon$	Constants required by Lemma 4.21.
$\lambda_{\theta,v}$	Discount coefficient of root-cause θ, used by ECED when computing Δ_{ECED}.
η	Parameter of f_{AUX} (see Equation (4.5.3), Lemma 4.21). It is only used for analysis.
λ	Parameter controlling the error rate of tests (see §4.6).
$r(\theta)$	$\text{supp}(\Theta) \to \mathcal{Y}$: mapping from root-causes $\theta \in \text{supp}(\Theta)$ to regions $y \in \mathcal{Y}$.
$\mathcal{R}_y$	Decision region indexed by target/decision y.
δ	The objective tolerance of prediction error.

Notations specific to §5

`cls`	Clustering function for evidence-candidate pairs.

γ	Discount factor for negative edges.
σ	Voting elements; evidence.
Σ	Set of voting elements.

Notations specific to §6

$\textsc{gap}_k$	Adaptivity gap of length k.
$\textsc{opt}_{const}(k)$	Optimal constant expected reward of f under cardinality constraint k.
$\textsc{opt}_{seq}(k)$	Optimal sequential expected reward of f under cardinality constraint k.
$\pi^*_{\text{batch},k}$	An optimal batch-mode policy (of fixed size).
π^*_{const}	An optimal constant policy.
π^*_{seq}	An optimal sequential policy.

Notations specific to §7

$\tilde{\Delta}$	Regret of an epoch for online adaptive information acquisition.
τ	Length of an epoch for online learning; horizon of an episode.
ϱ_{ij}	Conditional probability $\mathbb{P}\left[x_i = 1 \mid \theta_j\right]$ (§4, §7).

Part I

Introduction and Background

1

Introduction and Overview

We are awash in a deluge of data today. A massive amount of information is being generated at unprecedented scale in social, financial and scientific applications [Agr+12], and has been utilized to drive nearly every aspect of our modern society. While the promise of having the data is significant for most machine learning applications, there are many technical and ethical issues that must be addressed to fully leverage its potential. A crucial bottleneck for data analysis is the *limited capability* of accessing and processing the existing data, due to bandwidth, power, budget, computational, or legal constraints. Besides, much of this data might not be useful, and a small subset can be sufficient for making decisions. This leads to the following fundamental question in artificial intelligence:

How should we efficiently acquire the most useful information for decision making, when we are given limited resources?

As illustrated in Fig. 1.1, the optimal information acquisition problem has been extensively studied in many areas, including optimal experimental design, machine learning (active learning in particular), decision theory, operations research, multi-agent systems, sensor networks, and robotics. In many of these problems, one needs to adaptively make a sequence of decisions on which information to acquire next, taking into account the information collected in previous rounds.

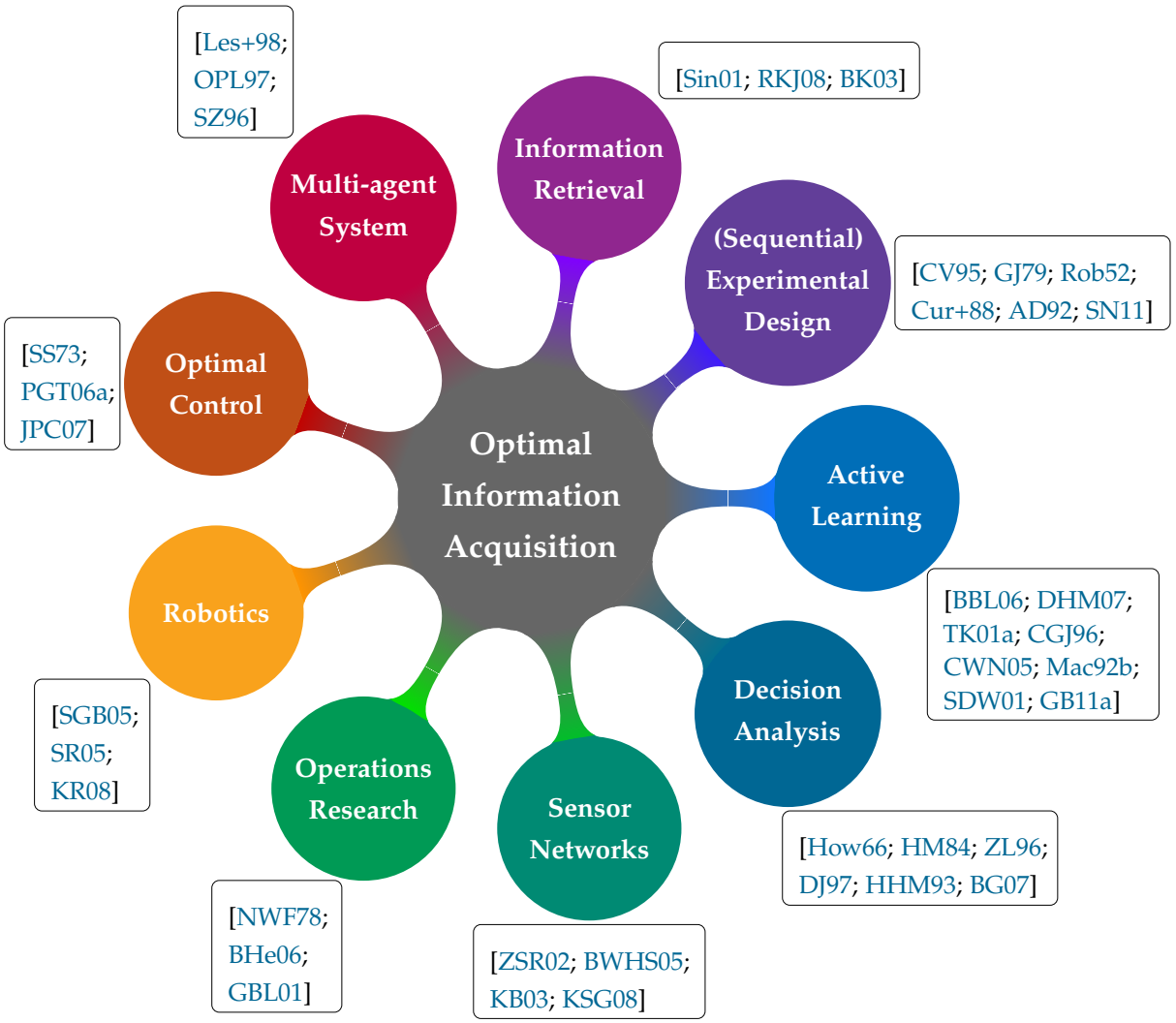


Figure 1.1: Some application domains of the optimal information acquisition problem.

Finding the optimal solution for information acquisition requires solving challenging stochastic optimization problems. In most cases, the objective functions for these problems are highly non-convex, and many of them are provably computationally intractable [LGM98]. Among existing techniques for solving such problems, most of them are either *heuristics approaches* which do not strive for optimality (e.g., uncertain sampling), or *principled* approaches that are difficult to scale to larger problems (e.g., non-myopic approaches for solving probabilistic planning problems). Both types of solutions could be problematic in practice. Therefore, it becomes of crucial importance to identify a class of such sequential decision problems, where we can develop tractable approximations of the optimal solution.

This dissertation pursues the fundamentals of adaptive information acquisition, with the goal to devise a mathematical and algorithmic framework for *efficient* (in terms of computational complexity) and *robust* (in terms of decision quality and noise-tolerance) decision making under uncertainty. Motivated by the demands of applications, it addresses the following five questions:

1. Due to insufficient knowledge about the environment and measurement noise, *uncertainty* is often an inseparable part of our models and observations. In the presence of uncertain observations, how can we characterize the trade-offs between reliability and efficiency?

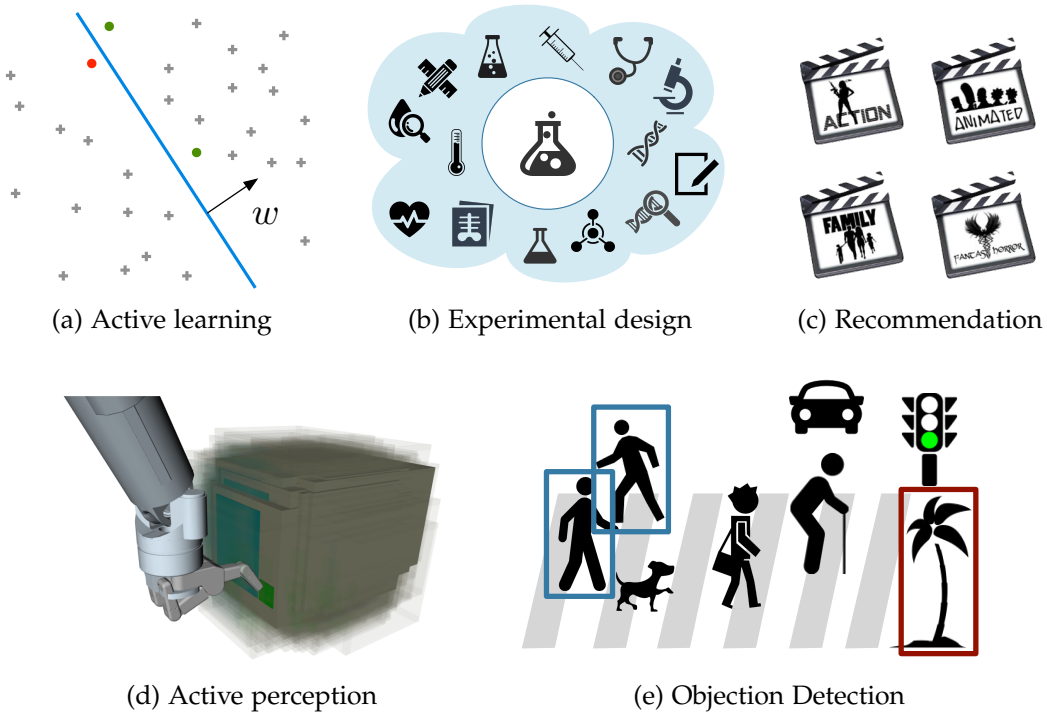
2. The true value of data lies not just in having it, but in harvesting it for making effective decisions. How can we design computationally tractable objectives for large-scale *decision problems* under partial observability?

3. Existing approaches for adaptive information acquisition are inherently sequential: One label is observed after the other. Is it possible to parallelize this process (e.g., having access to multiple labeling resources)? How can we exploit *information parallelism* for adaptive decision making?

4. In practice, the model that defines the objective function may be unknown to the algorithm and may have to be estimated *online* through experimentation. When the parameters of the model is not specified, how can we trade off acquiring information that is useful for learning the model (exploration), and acquiring information that is useful for making decisions (exploitation)?

5. While one focus of this dissertation is to devise fundamentally new algorithms, an equally important component is to demonstrate the effectiveness of the proposed algorithms and models in real-world applications. Which classes of applications are suited for adaptive decision making?

The above questions are stated in a fairly abstract manner. In the following, we will elaborate on theses questions through a few concrete examples. Afterwards we will state the main contributions of this dissertation.



(a) Active learning (b) Experimental design (c) Recommendation

(d) Active perception (e) Objection Detection

Figure 1.2: Examples of the Optimal Value of Information problem

1.1 Optimizing Value of Information

Let us consider the following applications:

- **Active learning.** *Active learning* [Set12; DL11] is a machine learning paradigm that allows active acquisition of training data (in terms of labeled *data instances*, or labeled *features*). For example, let us consider the *active instance labeling* problem: Given a set of unlabeled data instances, and some prior distribution on the underlying classifier, the goal is to learn a classifier by carefully requesting as few labels as possible. In the example shown in Fig. 1.2a, if we want to learn a linear separator to distinguish the two classes in the 2-D plane, we may only need to request the three highlighted examples in order to infer the correct decision boundary.

- **Sequential experimental design and active testing.** Experimental design [Lin56] aims at predicting the outcome of some phenomenon, by performing a series of carefully designed, informative experiments (See Fig. 1.2b). A typical application

domain of experimental design is medical and clinical research, where trails and experiments are performed to assess the safety and efficacy of some new product (e.g., new drug or device) in development. One can also think of *medical diagnosis* [Kon01; Ber+10], where a doctor can choose from a large set of medical tests, and she aims to administer a small subset of tests on the patient that will enable her to provide effective treatment. Similarly, in *troubleshooting* [HBR94], there are many tests one can perform on the system in question, and the goal is to run a small set of tests that provide sufficient information for repairing the system.

- **Active recommendation.** The goal of recommender systems is to predict the preference of a user based on user's profile (e.g., social profile, browsing history, etc.). In movie recommendation (See Fig. 1.2c), for example, the system aims to suggest a movie from a certain genre of interest, based on the user's ratings of her previously watched movies. To get enough information for an accurate prediction, the system may propose movies to the user for feedback, e.g., by explicitly asking her ratings of the movies, or by proposing pairs of movies to the user and asking which one she prefers [GLS08].

- **Active perception.** A common task in robotic manipulation is active localization, i.e., to localize an object by exploring the environment via some sensing actions, such as *touch* or *vision* [FBT98]. Imagine the task of pushing a button with the finger of a robotic end effector (See Fig. 1.2d). The robot can probe a location through *guarded moves* [WG75], where the end effector moves along a path until a contact is sensed. After sensing contact, the robot gets to know that there are certain object locations which could not have produced the contact, e.g. if they are far away. Here the goal is to design a sequence of probing actions so that it can gather enough information to guarantee success in pushing the button.

- **Active detection.** Efficient detection of multiple object instances is one of the fundamental challenges in computer vision. One way to enhance the performance of existing object detection algorithms is to have a (human) expert in the loop to provide feedback on the detection results, e.g., whether/ how the detection should be improved. To make the best use of such labeling resources, one needs to decide when to invoke the expert, such that the best possible performance can be achieved while requiring a minimum amount of supervision.

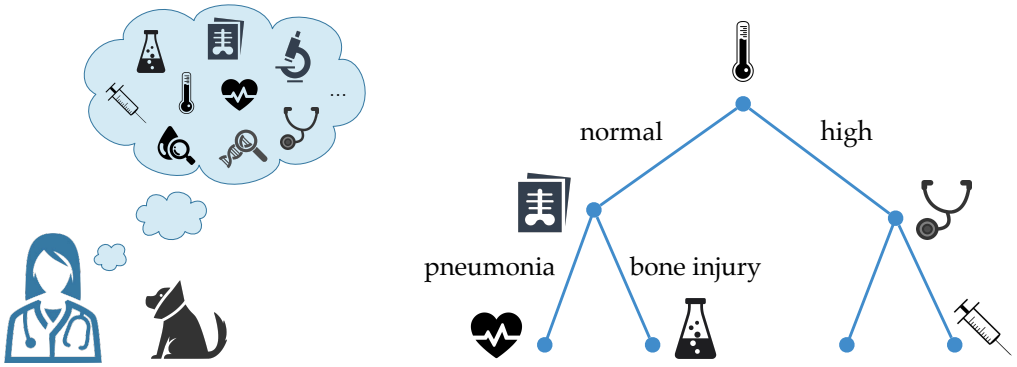
In all above scenarios, we wish to acquire a "useful" subset from a collection of examples, or *tests*, based on some objective that quantifies the *value* of a subset. We call the general class of information acquisition problems the *optimal value of information* problem. In the following, we introduce several different settings of the problem and present our contributions.

1.1.1 Sequential Information Maximization

Suppose we want to acquire information about some target hypothesis which is encoded by random variable Y. Intuitively, we want to perform a sequence of tests that are most *informative* about Y. This setting naturally maps to applications such as medical diagnosis (performing medical tests that are most informative about the patient's condition), recommendation (proposing a set of pair-wise comparison queries to identify the most relevant movie), active learning (selecting informative data instances to query, e.g., for learning a classifier), and numerous others.

A priori vs. sequential selection Maximizing the *informativeness* of a set of random variables (i.e., test outcomes) has a rich history in machine learning [Lut85; Mac92a]. It is perhaps best understood in the *a priori selection* setting, where the set of all tests to be executed is determined ahead of time, i.e., before any observations have been made. It is known that the problem of selecting a set of most informative tests of restricted cardinality is in general NP-hard [KLQ95]. By leveraging the theory of *submodular functions* [NWF78], Krause and Guestrin [KG05] showed that under some conditions, near-optimal solutions can be identified efficiently. In particular, under the assumption that the test outcomes are conditionally independent given the target variable Y, the informativeness of a set of tests can be modeled as a submodular function, and therefore a simple greedy algorithm leads to a $1 - 1/e$ approximation of the optimal solution.

In many applications, however, it is more natural to consider *sequential* selection. For example, in medical diagnosis, it is natural for a doctor to conduct a medical test based on the results of all previous tests (See Fig. 1.3). Comparing with the a priori selection setting where we specify all tests in advance (an *open-loop* selection), a sequential (*close-loop* selection) algorithm chooses a possibly different set of tests depending on the outcomes of the tests. Such adaptation provides an informational advantage, allowing to obtain more information than committing to all tests ahead of time.



(a) Medical diagnosis (b) Adaptive strategy for performing tests

Figure 1.3: Illustration of a sequential selection strategy for medical diagnosis.

The most informative selection criterion. A natural approach that finds widespread use in practice is the *Most Informative Selection* (MIS) criterion which, in each step, greedily picks the test that provides the maximal reduction in uncertainty (quantified in terms of Shannon entropy [Sha48]) about the target variable Y, conditioned on the observations gathered so far. Despite its widespread use, not much is known about the theoretical properties of this greedy algorithm, in particular in the practically important setting where observations are noisy. With this, a key question we want to answer is:

*When does the **most informative selection** criterion perform (provably) well?*

A general framework to study the performance of greedy algorithms is *adaptive submodularity* [GK11a]. It is known that if a sequential problem is adaptive submodular, then greedily optimizing it results in near-optimal performance. Unfortunately, the MIS criterion violates the adaptive submodularity condition and hence does not fall within this framework. In the case where tests are *noise-free* (i.e., their outcomes are deterministic functions of Y), it is known that greedily optimizing mutual information is effective [Das05b; ZRB05]. Under the noisy setting, however, the theoretical performance of the most informative selection criterion is much less well understood.

In §3 of this dissertation, we present the first rigorous information-theoretic analysis of the MIS criterion that holds even under persistent noise. Specifically, we consider a general probabilistic model, where the originally deterministic tests are corrupted by some arbitrary noisy channel. We derive a lower bound on the reward achieved by MIS in terms of a *channel separability condition*, a simple measure that characterizes the

severity of noise. We further provide an example to show that such measure is important in the bound. It follows from our results that under common assumptions made about the noise (e.g., symmetric flips of the binary outcome), the sequential information maximization criterion behaves near-optimally. Hence our results theoretically justify why the mutual information criterion has been found to be effective in these settings. Our analysis also sheds light on cases where greedy information maximization may fail, and thus nonmyopic policies, e.g., using look-ahead, might be required.

1.1.2 Information Gathering for Decision Making

While acquiring the "most informative" tests is useful, say, for reducing the uncertainty of some phenomenon of interest, in many applications collecting information is not the goal of its own, but rather a prerequisite for *making informed decisions*. In §4 of this dissertation, we aim to explore the question:

> *How can we adaptively acquire information, so that we can make **effective decisions**?*

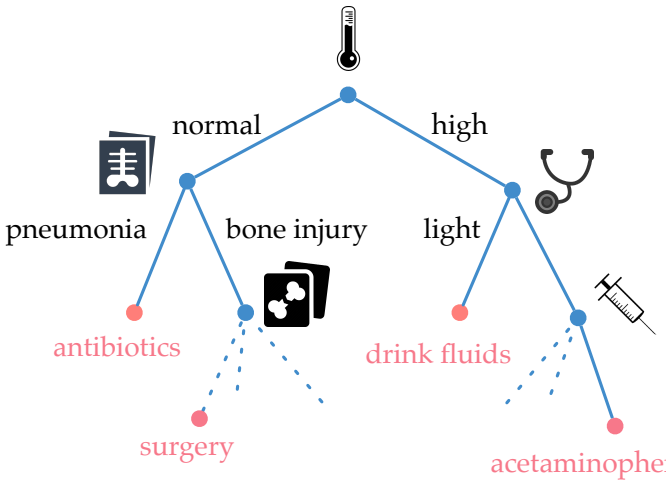


Figure 1.4: Illustration of a two-stage decision-making strategy.

Decision making under uncertainty can be viewed as a two-stage problem where in the first stage information is gathered, and in the second stage decisions are made based on the acquired information. We illustrate such processes with the medical diagnosis example in Fig. 1.4: a doctor first performs a sequence of medical tests to gather enough information about the symptoms, and in the end decides how to treat the patient.

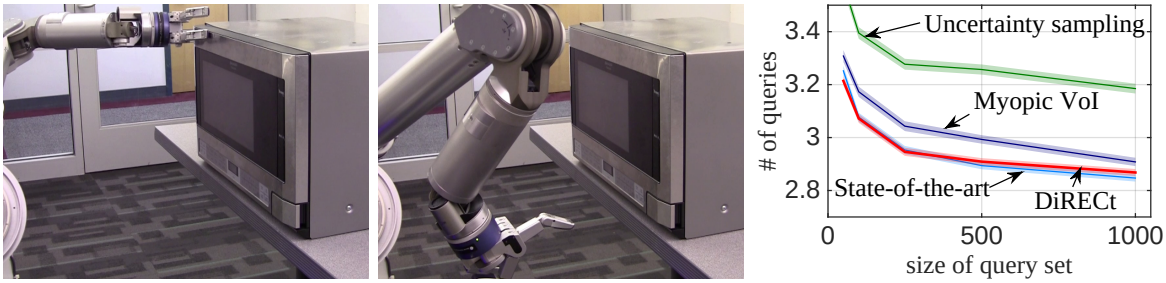


Figure 1.5: Efficient active touch-based localization. DiRECt achieves the state-of-the-art performance, but is more than five times faster.

Submodular surrogates for value of information. In contrast to objective functions for *estimation* (such as reduction in Shannon entropy), the value of information objective for *decision making* is typically *highly non-submodular*: Often, we may require multiple observations to even consider changing our decision, violating the diminishing returns condition. In fact, for optimizing value of information in general, standard greedy algorithms can perform arbitrarily poorly [GKR10a].

In this dissertation, we demonstrate that it is possible to efficiently construct a *surrogate objective function* for the optimal value of information problem. In particular, we propose an equivalent formulation, which we call the *Decision Region Determination* (DRD) problem, and provide a principled framework for addressing it. Our algorithm, DiRECt, relies on a surrogate objective function which is *adaptive submodular*, and can therefore be greedily optimized. It is designed in a way that optimizing the surrogate provably leads to improvements in the decision performance. We evaluate our algorithm on four applications: a touch-based localization application on a real robot platform [Jav+13], a Bayesian experimental design task intended to distinguish among economic theories of how people make risky decisions [Ray+12], an active preference learning task via pairwise comparisons [KIM12], and an adaptive management task for biodiversity conservation [RCL11]. Our experimental results show that DiRECt significantly outperforms myopic value of information in most settings (see Fig. 1.5 to preview our result on the touch-based localization application).

Efficient optimization of value of information. While our submodular surrogates-based approaches show promising performance in several applications, it can still be computationally challenging when deployed in real-world scenarios. In many data-intensive decision-making applications, evaluating these surrogate objectives can be

expensive: At each iteration, one needs to perform a greedy search over the tests and find the one that myopically maximizes the expected gain in the corresponding objective, whose runtime depends linearly on the size of the support of the probability distribution over the outcomes of all tests. With the size of the support growing exponentially in the number of tests, it is often computationally prohibitive to work with the original distribution. In this dissertation, we employ a dynamic hypothesis enumeration strategy and show that with sufficient amount of samples, one can identify a near-optimal decision with high probability. Our approach, comparing to prior sampled-based work, leverages the structure of the probabilistic model, and thereby offers increased efficiency and better approximation guarantees. Our empirical results show that one can efficiently run the submodular surrogate-based approaches with our hypothesis enumeration strategy, while achieving much better performance comparing with commonly-used heuristics (we observe a 16% improvement in terms of the average cost on a real-world online troubleshooting platform).

Dealing with i.i.d. noise. The near-optimal performance guarantee of DiRECt relies on the assumption that the observed outcomes of selected tests are *noise-free*. It means that, for example, in medical diagnosis, observing the outcome of a medical test would *rule out* the possibility of a particular disease. However, due to observation error in practical scenarios, a more realistic setting is to consider noisy tests.

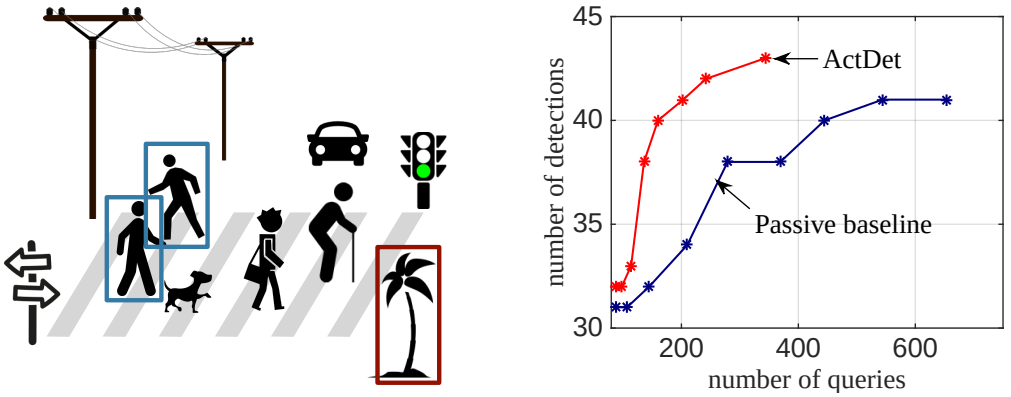
We introduce ECED, a novel algorithm for Bayesian active learning and decision making, and prove strong theoretical guarantees with *noisy test outcomes*. We demonstrate the compelling performance of ECED on two real-world problem instances, including comparison-based preference learning [KIM12] and Bayesian experimental design for preference elicitation in behavioral economics [Ray+12]. We prove that when the test outcomes are binary, and the noise on test outcomes are mutually independent, then ECED is guaranteed to obtain near-optimal cost. We develop a theoretical framework for analyzing such sequential policies. The key insight is to show that ECED is effectively making progress in the long run as it picks more tests, even if the myopic choices of tests do not have immediate gain in terms of reducing the uncertainty of the target variable.

1.1.3 Other Applications

In the previous discussion, we explained that it is possible to construct a surrogate function for the decision-theoretic value of information problem. Moving beyond the two-stage process (i.e., in the first stage we collect information; in the second stage we make a final decision based on the information acquired), a key question that we are interested in is:

> *Can **submodular surrogate functions** be developed for solving more general sequential decision problems?*

We show that the technique of constructing submodular surrogates for non-submodular problems is generally useful for adaptive information acquisition problems. In §5, we focus on the *active detection* application (see Fig. 1.6a). Different from the two-stage decision-making process, in this problem, the phases of information gathering (e.g., querying candidate detection to get useful feedback) and decision making (e.g., proposing candidates that are most likely to be an actual object) are interleaved. Hence, it requires us to trade off gathering information and maximizing utility.



(a) Sequential detection of multiple objects

(b) Results on biodiversity monitoring

Figure 1.6: Sequential object detection: which location to query next?

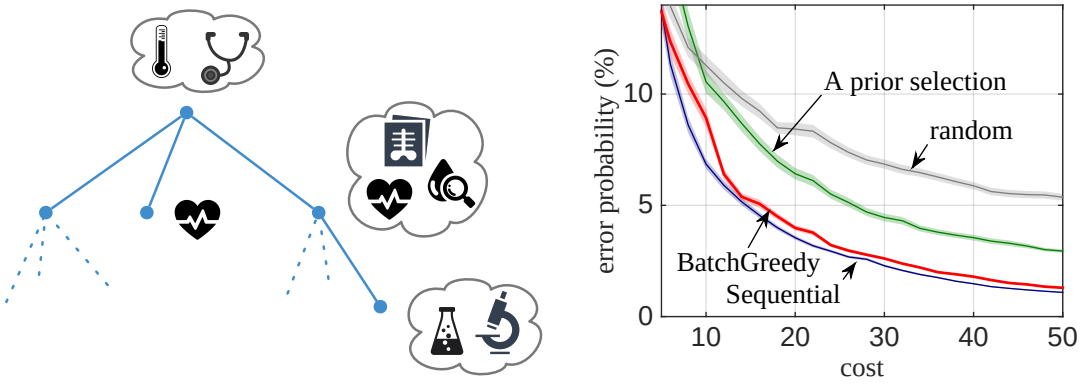
We propose a principled approach to the active object detection problem, and show that for a rich class of base detection algorithms, one can derive a natural sequential decision problem for deciding when to invoke expert supervision. We further show that the objective function satisfies adaptive submodularity, which allows us to obtain strong performance guarantees for our algorithm. We demonstrate the performance of

the proposed algorithm on three real-world tasks, including a problem for biodiversity monitoring from micro UAVs in the Sumatra rainforest. In Fig. 1.6b we show a preview of our result for this biodiversity monitoring application. Our results show that active detection not only outperforms its passive counterpart; for certain tasks, it also works significantly better than the straightforward application of existing active learning techniques.

1.2 Practical Challenges

1.2.1 Exploiting Information Parallelism

We have considered adaptive information acquisition as a sequential process, where one test is picked at each iteration, and the next test cannot be picked until the outcome of the previous test is observed. In many practical settings, however, fully sequential selection, where the choice of the next example depends on all previous labels, is infeasible. For example, when recruiting workers on Amazon Mechanical Turk for crowdsourcing annotation, one usually generates tasks comprising several unlabeled examples. Similarly, in high-throughput experimental design, it is often more cost-effective to perform several experiments in parallel (See Fig. 1.7a).



(a) Batch selection of tests

(b) Results: Batch-mode Active Learning

Figure 1.7: Exploiting information parallelism.

More generally, in many sequential decision problems, we would like to choose multiple actions to be performed in parallel and receive feedback only after all actions have been

carried out. This feedback then informs the next batch of operations. For instance, in the viral marketing application described in §1.1, it is natural to conduct a multi-stage marketing campaign, where each stage is informed by the observed effectiveness of the previous stage. Similar problems arise in resource allocation in computational sustainability [Gol+11], and vaccination problems in epidemiology [Ans+09]. These practical challenges motivate us to investigate the following question:

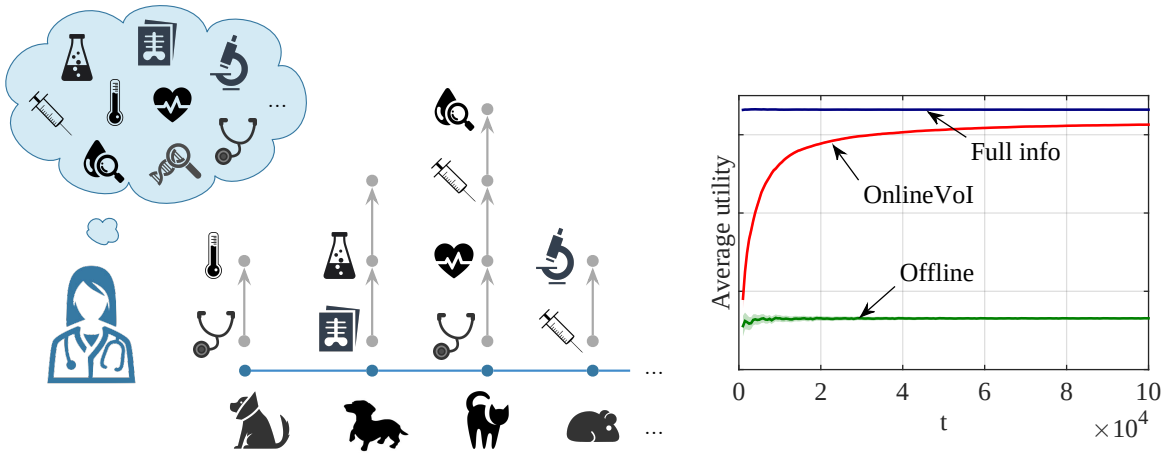
*How can we exploit **information parallelism** for adaptive decision making?*

In §6 of this dissertation, we study *information-parallel learning and decision making*. In particular, we tackle batch-mode active learning and more general stochastic optimization problems, such as influence maximization in social networks, that exhibit *adaptive submodularity* [GK11b]. We prove that, for such problems, a simple BatchGreedy approach, which greedily selects examples within a batch, and assembles batches in a greedy manner, is competitive with the optimal batch-mode algorithm. Furthermore, we prove that perhaps surprisingly, in some natural settings, the *price of parallelism* is bounded: the use of batches incurs competitively low cost *irrespective* of the batch size, even when compared to a *fully sequential* policy. We demonstrate the effectiveness of our approach on active learning tasks, as well as adaptive influence maximization in social networks. Fig. 1.7b shows the performance of BatchGreedy on an active learning task, where each batch contains ten tests. Our approach is the first to provide both strong guarantees and compelling empirical performance for the important practical problem of batch mode active learning, where BatchGreedy improves on random selection by $\approx 48\%$ more than state of the art does on our test sets.

1.2.2 Unknown Parameters: Converting Offline to Online

In the previous discussion, we assume that the objective functions defining the value of information are either explicitly given (e.g., the MIS criterion), or constructed through some (deterministic) algorithmic framework (e.g., the DiRECt objective). In either case, the objective is given in terms of a probabilistic model estimated from historical data, according to which the reward function can be computed. However, in practice, such a model may be unavailable and may have to be estimated through experimentation (See Fig. 1.8a).

In §7 of this dissertation, we aim to answer the following question:



(a) Online optimization of VoI

(b) Results on *Troubleshooting*

Figure 1.8: Converting offline to online.

How can we solve the sequential decision problems,
if we have to learn the model ***online****?*

We investigate the *online* sequential information acquisition problem, where parameters of the probabilistic model are initially unknown, and can only be learned from data in an online fashion. For instance, in troubleshooting the conditional probabilities of symptoms given a root-cause might be unknown. For this purpose, we integrate the offline algorithms considered in §3 and §4 into an online learning framework, OnlineVoI, by employing a *posterior sampling* approach. We establish a rigorous bound on the *expected regret* (defined in terms of the value of information) of our framework. Finally, we demonstrate our online learning framework on a real-world troubleshooting platform. Our experiments under the online setting imply that our framework encourages efficient exploration, which, combined with the sampling algorithm, leads to effective online learning of the optimal VoI (See Fig. 1.8b to preview our result on an online troubleshooting application).

1.3 Organization of this Dissertation

We summarize the key contributions of this dissertation in Table 1.1.

In Part I of this dissertation, we will provide background on the optimal information

<table>
<tr><td rowspan="2">Part I</td><td rowspan="2">Settings

Applications</td><td colspan="2">Decision-theoretical VoI</td><td rowspan="2">General notion of “coverage”</td></tr>
<tr><td>Inf. Gathering</td><td>Decision Making</td></tr>
<tr><td colspan="2">experimental design, recommendation, active learning, localization, etc.</td><td>viral marketing object detection</td></tr>
<tr><td>Part II</td><td>Sequential</td><td rowspan="2">MIS (§3)</td><td>DiRECt (§4)</td><td rowspan="2">ActDet (§5)</td></tr>
<tr><td></td><td>ECED (§4)</td></tr>
<tr><td rowspan="2">Part III</td><td rowspan="2">Batch-mode
Online</td><td colspan="3">BatchGreedy (§6)</td></tr>
<tr><td colspan="3">OnlineVoI (§7)</td></tr>
</table>

Table 1.1: Summary of key contributions.

acquisition problem. We will also review existing work that is relevant to most chapters of this dissertation; in particular, we will survey the background on active learning, review the concept of submodularity, and review existing results on (adaptive) submodular optimization (§2). More specific discussions of work related are presented in the subsequent chapters.

In Part II, we will then elaborate on our approaches for the optimizing value of information for the adaptive information acquisition problem, including problems of sequential information maximization (§3), information gathering for decision making (§4) and active object detection (§5). We also report results obtained by applying existing algorithms for the adaptive information acquisition problems considered in this dissertation.

In Part III, we investigate algorithms addressing the practical challenges. In §6, we develop algorithms for batch selection (i.e., parallelization) of the adaptive stochastic optimization problem. In §7, we look into settings where parameters of the models are initially unknowns, and propose an online learning framework for optimizing value of information.

Lastly, in Part IV we present our conclusions of this dissertation (§8) and propose a few interesting directions for future work.

1.4 Publications Relevant to this Dissertation

This dissertation is based on materials from the following conference publications and technical reports.

- Yuxin Chen, S. Hamed Hassani, Amin Karbasi, and Andreas Krause. "Sequential Information Maximization: When is Greedy Near-optimal?" In: *Proc. International Conference on Learning Theory (COLT)*. 2015 (§3)
- Yuxin Chen, Shervin Javdani, Amin Karbasi, James Andrew Bagnell, Siddhartha Srinivasa, and Andreas Krause. "Submodular Surrogates for Value of Information". In: *Proc. Conference on Artificial Intelligence (AAAI)*. 2015 (§4)
- Yuxin Chen, S. Hamed Hassani, and Andreas Krause. "Near-optimal Bayesian Active Learning with Correlated and Noisy Tests". In: *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2017 (§4)
- Yuxin Chen, Hiroaki Shioi, Cesar Fuentes Montesinos, Lian Pin Koh, Serge Wich, and Andreas Krause. "Active Detection via Adaptive Submodularity". In: *Proc. International Conference on Machine Learning (ICML)*. 2014 (§5)
- Yuxin Chen and Andreas Krause. "Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization". In: *International Conference on Machine Learning (ICML)*. 2013 (§6)
- Yuxin Chen, Jean-Michel Renders, Morteza H. Chehreghani, and Andreas Krause. *Efficient Online Learning for Optimizing Value of Information: Theory and Application to Interactive Troubleshooting*. Tech. rep. 2016 (§4, §7)

The following publication is also relevant to this dissertation. In particular, in the following publication we introduced the HEC algorithm, which is used as a baseline algorithm for the evaluation of DIRECT. We briefly discuss the HEC algorithm in §4.2.2.

- S. Javdani, Y. Chen, A. Karbasi, A. Krause, D. Bagnell, and S. Srinivasa. "Near-Optimal Bayesian Active Learning for Decision Making". In: *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2014

1.5 Collaborations

This dissertation will be much different without the valuable inputs of my collaborators. Hereby, I mention the researchers involved in the work presented in this dissertation. Evidently, my advisor, Prof. Andreas Krause was actively involved in all the work presented in this dissertation. The research presented in §3 and §4.5 was done in close collaboration with Hamed Hassani, who offered invaluable inputs throughout the projects, which turned out to play a key role in devising the theoretical framework for analyzing MIS and ECED. Amin Karbasi actively participated in the numerous meetings and discussions on the optimal value of information presented in §4.1.1, which eventually resulted in the DIRECT framework. Shervin Javdani and I started working on the DRD problem (§4.1.1) in parallel, and collaboratively we developed HEC and DIRECT. Shervin Javdani conducted the active localization experiment in §4.6.5. He was supervised by Prof. Drew Bagnell and Prof. Siddhartha Srinivasa from the Robotics Institute at Carnegie Mellon University, who also provided valuable input for this project. The dynamic hypothesis enumeration framework (§4.4) and the online learning framework (§7) is developed in close collaboration with Jean-Michel Renders and Morteza Haghir Chehreghani while I was visiting Xerox Research Center Europe. Hiroaki Shioi conducted experiments on the orangutan nests detection project (§5.3.1) as part of his Master's thesis, and Cesar Fuentes Montesinos conducted experiments on the PASCAL VOC 2008 dataset for person detection (§5.3.3) as part of his Master's thesis. Both Hiroaki and Cesar were supervised in their theses by Prof. Andreas Krause and me. The orangutan nests images were provided by Prof. Lian Pin Koh and Prof. Serge Wich, who also participated in discussions about the project. The results on the adaptivity gap presented in §6.2.2 are based on unpublished work in collaboration with Gábor Bartók, who provided the proofs of Theorem 6.6 and Theorem 6.7 on the adaptivity gap.

2

Background and Related Work

As a fundamental component in many application domains in artificial intelligence, the optimal information acquisition problem has been studied in a vast extent of literature. Roughly speaking, these approaches vary in how the *usefulness* of information is defined and which optimization techniques are used. There are (i) *Frequentist* approaches, such as classical optimal experimental design [AD92; BV04; DA88; Puk93] and most active learning techniques [BBL06; Das05a], and (ii) *Bayesian* approaches, such as Bayesian experimental design [Cur+88; CV95; SW97; SN11] and most decision theoretic techniques [How66; HHM93]. In comparison with Frequentist approaches, Bayesian approaches are in general more expressive: it allows to utilize rich probabilistic models [KF08], by making assumptions about the likelihood of particular observations. As a result, one can plug in generic objective functions to application specific models. In this dissertation, we adopt the Bayesian perspective due to its expressiveness.

In this chapter, we first state the general class of adaptive information acquisition problem (§2.1) studied in this dissertation, followed by introducing a few candidate reward functions (§2.2) relevant to most chapters. We then provide a survey of the related work, in particular on the optimization techniques (i.e., greedy approaches and adaptive submodularity (§2.3), non-myopic approaches such as probabilistic planning (§2.4)), as well as connections to active learning (§2.5) and other related areas (§2.6). We will review additional related work that is specific to different problem settings in the subsequent chapters.

2.1 The Adaptive Information Acquisition Problem

We begin by introducing the basic notations and terminology, and then formally state the general class of adaptive information acquisition problems.

2.1.1 Tests, Outcomes, and Policies

Suppose we are given a finite set of *tests* $\mathcal{V} = \{1, \ldots, t\}$ (e.g., medical tests, troubleshooting questions, nodes in a social network, or unlabeled data points), and we wish to select a subset $\mathcal{A} \subseteq \mathcal{V}$ among those tests. Each test $v \in \mathcal{V}$ is associated with some *observable* random variable $X_v \in \mathcal{X}$, where $\mathcal{X}$ is the set of all possible *outcomes* of any test. For example, X_v may correspond to the result of a medical test, the symptom of a malfunctioning system, the outcome of a probing action, the number of people one can influence in a social network, or the label of a data point. Let x_v denote the observed outcome of X_v. We use $\mathbf{X}_\mathcal{V}$ to denote the collection of random variables $\{X_1, \ldots, X_t\}$, and use $\mathbf{x}_\mathcal{V}$ to denote their observed outcomes $\{x_1, \ldots, x_t\}$. Similarly, let $\mathcal{A}$ be a subset of the ground set $\mathcal{V}$, we use $\mathbf{X}_\mathcal{A}$ to denote the set of random variables indexed by the elements of $\mathcal{A}$, and use $\mathbf{x}_\mathcal{A}$ to denote a *partial realization* of the tests.

We assume that the joint probability distribution $\mathbb{P}[\mathbf{X}_\mathcal{V}]$ over theses random variables is defined in a *probabilistic model*. Therefore, after making partial observations $\mathbf{x}_\mathcal{A}$, we can compute a posterior distribution $\mathbb{P}[\mathbf{X}_\mathcal{V} \mid \mathbf{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}]$ through *Bayesian inference* (e.g., by exact inference [LS88], or Markov Chain Monte Carlo [Met+53; GG84], etc.). This posterior distribution characterizes the uncertainty about the state of the environment, which we can use to make decisions.

In this dissertation, we consider *adaptive strategies* for picking the tests. For now, let us consider problems where we *sequentially* pick test a $v \in \mathcal{V}$, get to observe its associated outcome x_v, pick the next test based on the previous observations, get to see its outcome, and so on. Later in Part III we will consider more general strategies where we may pick tests in batches. We encode such an adaptive strategy as a (sequential) *policy* π. Formally, a policy $\pi : 2^{\mathcal{V} \times \mathcal{X}} \nrightarrow \mathcal{V}$ is defined to be a *partial mapping*[1] from partial realizations (e.g., observation vector $\mathbf{x}_\mathcal{A}$) to tests, specifying which test to run next (or that we should stop testing if $\mathbf{x}_\mathcal{A}$ is not in the domain of π). Each deterministic policy

[1] A partial function $f : X \nrightarrow Y$ is a function $f : X' \to Y$ for some subset X' of X.

can naturally be associated with a decision tree (see Fig. 2.1). We use the notation $\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})$ to refer to the set of tests selected by π under realization $\mathbf{x}_\mathcal{V}$.

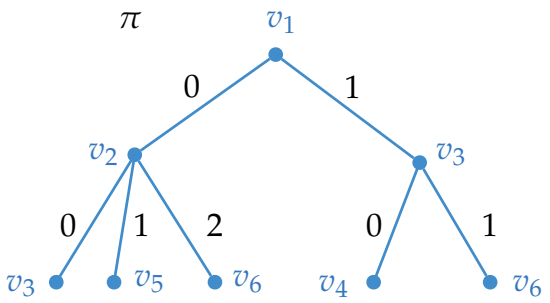


Figure 2.1: Illustration of an adaptive (deterministic) policy via a decision tree. Nodes represent tests picked by the policy, and edges represent the outcomes of tests.

2.1.2 Reward and Cost

The usefulness or quality of the observed outcomes $\mathbf{x}_\mathcal{A}$ is defined through some reward function $f : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$, where $f(\mathbf{x}_\mathcal{A})$ depends on the realization of the random variable $\mathbf{x}_\mathcal{V}$, and the chosen set of tests $\mathcal{A}$. The choice of the reward function may vary in different applications. Before deploying a policy π, the particular realizations $\mathbf{X}_\mathcal{V} = \mathbf{x}_\mathcal{V}$ are not known in advance. Therefore, it is natural to define a reward function F that quantifies the expected value,

$$F(\pi) = \mathbb{E}_{\mathbf{x}_\mathcal{V}}[f(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V}))] = \sum_{\mathbf{x}_\mathcal{V}} \mathbb{P}\left[\mathbf{x}_\mathcal{V}\right] f(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) \tag{2.1.1}$$

where the expectation is taken over all the possible realizations $\mathbf{X}_\mathcal{V} = \mathbf{x}_\mathcal{V}$.

In practice, tests are costly (e.g., there is a cost for setting up the medical diagnosis/troubleshooting platform, for promoting a product, or for labeling a data point, etc.), and we only have limited budget and resources to spend on conducting the tests. In principle, one can incorporate the cost directly into the reward; alternatively, we choose to model the cost into a separate set function $C : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$, mapping sets of observations to (non-negative) real numbers. Unless explicitly pointed, in this dissertation we mainly consider *additive cost* functions, i.e., we assign a cost $c(v)$ for each test $v \in \mathcal{V}$, and define the cost of a subset $C(\mathbf{x}_\mathcal{A})$ as $C(\mathbf{x}_\mathcal{A}) = \sum_{v\in\mathcal{A}} c(v)$.

2.1.3 Adaptive Stochastic Maximization and Coverage

Now that we have defined *policy* and its *value*, let us look at the common structure of the adaptive information acquisition problem. There is a set of random variables, and we follow some adaptive policy to reveal a subset of those (because there is a cost). At some point, the policy decides to stop, and we observe a subset of observations, which determines the value (or expected reward) of the policy. Our general goal is to design a policy π^*, such that the expected reward F is maximized, while the cost of running the policy cost is minimized. This is a bi-criteria optimization problem [BV04]. One possibility is to maximize the average reward subject to some budget constraint:

$$\pi^* \in \arg\max_{\pi} F(\pi), \text{ subject to } C(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) \leq B \text{ wherever } \mathbb{P}[\mathbf{x}_\mathcal{V}] > 0. \tag{2.1.2}$$

Here, B is a budget on the maximal cost of the policy. In the viral marketing example, we would like to maximize the expected number of people influenced, by adaptively offering free products to the most influential set of people, while not exceeding budget B. In active learning, we would like to maximize the performance of the trained classifier (e.g., by minimizing the expected loss), by adaptively choosing a set of data points to label with labeling cost at most B. This problem is often referred to as the *Budgeted Adaptive Stochastic Maximization* problem [GK11a].

Alternatively, we can specify a quota Q of reward to achieve, and try to find the cheapest policy achieving the quota. We define the expected cost of a policy as the expected number of tests it picks, i.e., $\text{cost}_{avg}(\pi) = \mathbb{E}_{\mathbf{x}_\mathcal{V}}[C(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V}))]$. The problem is then to find

$$\pi^* \in \arg\min_{\pi} \text{cost}_{avg}(\pi), \text{ subject to } f(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) \geq Q \text{ wherever } \mathbb{P}[\mathbf{x}_\mathcal{V}] > 0, \tag{2.1.3}$$

In the medical diagnosis example, it is natural to think of minimizing the expected cost of performing medical tests, so that we can treat the patient properly. Similarly, in troubleshooting, we may want to minimize the cost of tests to fix the system in question. Such problem is called the *Adaptive Stochastic Minimum Cost Coverage* problem [GK11a]. Another variant of this problem is to minimize the worst-case cost $\text{cost}_{wc}(\pi) = \max_{\mathbf{x}_\mathcal{V}} C(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V}))$, i.e., the cost incurred under adversarially chosen realizations. Optimizing the worst-case cost may be useful for some applications, since it may be necessary to use a pessimistic analysis to guarantee safety.

2.2 Choice of Reward Functions

By choosing different forms for the reward function f, we can model very different classes of adaptive information acquisition problems. In this section, we introduce a few particular reward functions, which we will examine in more detail in Part II of this dissertation.

2.2.1 Mutual Information

Consider the task of learning the value of some unknown target random variable Y from a set of tests. A widely used notion of *informativeness* is given by the reduction of Shannon entropy [Sha48] about the target variable. In this case, the reward of selecting tests $\mathcal{A}$ and observing $\mathbf{x}_\mathcal{A}$ is

$$f(\mathbf{x}_\mathcal{A}) = \mathbb{H}(Y) - \mathbb{H}(Y \mid \mathbf{x}_\mathcal{A}), \tag{2.2.1}$$

where $\mathbb{H}(Y) = -\sum_y \mathbb{P}[Y = y] \log_2(\mathbb{P}[Y = y])$ denotes the Shannon entropy of the target variable Y under the prior distribution $\mathbb{P}[Y]$, and $\mathbb{H}(Y \mid \mathbf{x}_\mathcal{A})$ denotes the Shannon entropy of Y under the posterior distribution $\mathbb{P}[Y \mid \mathbf{x}_\mathcal{A}]$. This leads to the objective function

$$F(\pi) = \mathbb{H}(Y) - \mathbb{E}_{\mathbf{x}_\mathcal{V}}[\mathbb{H}(Y \mid \mathcal{S}(\pi, \mathbf{x}_\mathcal{V}))] = \mathbb{I}(Y; \pi). \tag{2.2.2}$$

Here, we use $\mathbb{I}(Y; \pi)$ to refer to the *mutual information* between our policy π and the target variable Y [CT91]. This objective leads to the most informative selection criterion, which will be examined in details in §3.

2.2.2 Decision-theoretic Value of Information

In many decision-making applications, the goal is to *act* upon the acquired information, rather than to minimize the uncertainty of some latent variable. A natural formalism for gathering decision-relevant information is the decision-theoretic notion of *value of information* [How66]. In this setting, there is some latent random variable Θ which encodes the relevant information for making a decision. In the medical diagnosis example, Θ would encode the presence or absence of a particular disease, or the physical condition of a patient. Upon observation of $\mathbf{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}$, a decision y is made

from a set of possible decisions $\mathcal{Y}$. For example, a decision could be a particular treatment for the patient. If the actual value of the hidden variable is $\Theta = \theta$, then this decision would result in a utility $u(\theta, y)$. Apparently, we wish to make an optimal decision $y \in \mathcal{Y}$ maximizing the utility $u(\theta, y)$. Since the hidden state Θ is unobserved, the best we can do is to make the decision that maximizes the *expected* utility

$$f(\mathbf{x}_{\mathcal{A}}) = \max_{y} \mathbb{E}_{\theta}[u(\theta, y) \mid \mathbf{x}_{\mathcal{A}}]\,. \tag{2.2.3}$$

This reward function $f(\mathbf{x}_{\mathcal{A}})$ is called the *decision-theoretic value of information*. Therefore, the (expected) value of information of a policy π is then defined as

$$F(\pi) = \mathbb{E}_{\mathbf{x}_{\mathcal{V}}}[f(\mathbf{x}_{\mathcal{A}})] = \sum_{\mathbf{x}_{\mathcal{V}}} \mathbb{P}\left[\mathbf{x}_{\mathcal{V}}\right] \max_{y} \mathbb{E}_{\theta}[u(\theta, y) \mid \mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}})]. \tag{2.2.4}$$

The goal of the optimal (decision-theoretic) value of information problem is then to find a cheapest policy π^* such that the decision-theoretic value of information is maximized. By fixing a quota on the value of information to be achieved, we can view this problem as an instance of the adaptive stochastic minimum cost coverage problem. We focus on this problem in §4.

2.2.3 Other Objectives

While we primarily focus on those posterior-based objectives (i.e., we use the observations $\mathbf{x}_{\mathcal{A}}$ to compute the posterior, and then score the posterior in some way), we also look at objectives where the value of the set of observations is quantified in other ways. In general, most of these objectives can be interpreted as some notion of *coverage*. For example, many applications have chosen ad hoc reward functions, such as the geometric notion of coverage for adaptive sensing [Hau+12], population coverage for information propagation in social networks [KKT03], etc. Besides these domain-specific approaches, there are statistical methods that use more *generic* objective functions, such as coverage (or reduction) of version space mass for active learning [TK01a; DKM05; GK11b], coverage of probabilistic votes for object detection [BLK12b], etc. We will provide more details on how one can utilize such objectives for the active object detection task in §5.

2.3 Greedy Approaches and Submodularity

Once the objective function has been chosen, different techniques can be applied for optimization. Perhaps the most commonly used approach in practice is to use a simple greedy policy. Often people rely on greedy *heuristics* without any theoretical guarantees [KHB07; DJ97]. However, it has been shown that there are a rich class adaptive stochastic optimization problems for which simple greedy solutions obtain near-optimal performance. Crucially, these objectives satisfy *adaptive submodularity* [GK11b], a natural diminishing returns condition as an adaptive analog of submodularity. We will review the relevant results for adaptive submodular optimization in this section.

2.3.1 Submodular Functions

Under the open-loop setting, many information acquisition problems as mentioned earlier satisfy an intuitive *diminishing returns* property: Performing $v \in \mathcal{V}$ helps more, if few tests have been performed, and less if many tests have already been performed. This intuition is formalized by the combinatorial notion of submodular functions.

Definition 2.1 (Submodularity [NWF78])**.** A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is called *submodular*, if for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $v \in \mathcal{V} \setminus \mathcal{B}$ it holds that

$$\Delta_f(v \mid \mathcal{A}) \geq \Delta_f(v \mid \mathcal{B}), \tag{2.3.1}$$

where for any set $\mathcal{A}'$,

$$\Delta_f(v \mid \mathcal{A}') = f(\mathcal{A}' \cup \{v\}) - f(\mathcal{A}')$$

is the *marginal benefit* of adding s to $\mathcal{A}'$.

We also define *monotonicity*, which characterizes the property that "performing additional test never hurts".

Definition 2.2 (Monotonicity)**.** A set function f is called *monotonic*, if for all $\mathcal{A} \subseteq \mathcal{V}$ and $v \in \mathcal{V}$ it holds that $\Delta_f(v \mid \mathcal{A}) \geq 0$.

Many information acquisition tasks can be cast as constrained submodular optimization problems. For instance, under the open-loop setting, the classical notion of submodularity has proven useful for active learning [Hoi+06a; GB11a]. In Hoi et al. [Hoi+06a],

both individual diversity and informativeness are evaluated w.r.t. the Fisher information (which is a submodular function). Guillory and Bilmes [GB11a] investigate active learning on graph-structured data with graph cut objective, which is monotonic submodular.

Examples of monotonic submodular functions also arise in many *physical sensing optimization* problems, where the goal may be to find the best k locations to place sensors in order to minimize the expected mean squared prediction error [DK08], or to maximize the expected reduction in Shannon entropy [KG09]. Other application domains include social networks analysis (e.g., selecting most informative blogs to cover the biggest stories propagating over the blogosphere [Les+07]), information retrieval (e.g., maximizing diversity of search results [YJ08], ad display for sponsored search [RKJ08]), document summarization [LB11], and so on.

Submodularity is a natural analog of convexity [Lov83]; for example, similar to minimization of convex functions, unconstrained submodular minimization is possible in polynomial time [GLS81]. Submodular *minimization* has been successfully applied in computer vision [VKR08], graphical model structure learning [CG07] and clustering [NJB05]. In contrast to convex maximization, which is generally very hard, one significant advantage of submodularity is the existence of strong approximation algorithms for *maximization*. One seminal result is due to Nemhauser, Wolsey, and Fisher [NWF78]: Suppose we would like to find an optimal set $\mathcal{A}$ of k observations (e.g., sensor locations) maximizing $f(\mathcal{A})$. Then the simple greedy algorithm, which iteratively adds the observation $v \in \mathcal{V} \setminus \mathcal{A}$ to the current set $\mathcal{A}$ that maximizes the reward $f(\mathcal{A} \cup \{v\})$ until k observations have been selected, achieves a constant fraction of $(1 - 1/e) \approx 63\%$ of the value attained by the optimal solution to this NP-hard optimization problem [NWF78]. In addition, submodularity can be exploited to speed up the greedy algorithm [Min78; Les+07], incorporate more complex constraints [KG07; NW81], and allow robustness [Kra+09] etc.

2.3.2 Adaptive Submodularity

The classical notion of submodularity is limited from a decision-theoretic standpoint, in a way that it requires us to commit to all decisions that we will make ahead of time, in an open-loop fashion. Interestingly, many objective functions arising in active learning problems satisfy *adaptive submodularity* (c.f. [GK11b]), a generalization of the classical

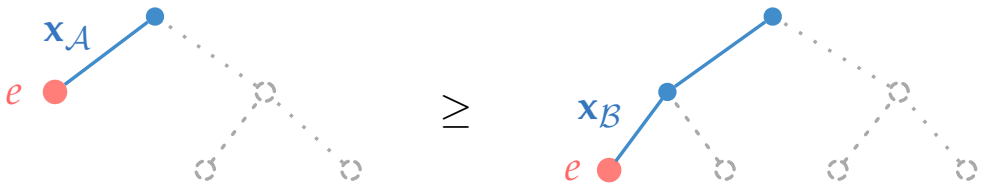


Figure 2.2: Adaptive Submodularity: for all $\mathbf{x}_\mathcal{A} \preceq \mathbf{x}_\mathcal{B}$ and $v \notin \mathcal{B}$ it holds that $\Delta_f(v \mid \mathbf{x}_\mathcal{A}) \geq \Delta_f(v \mid \mathbf{x}_\mathcal{B})$. In words, it implies that the gain of item v, in expectation over its unknown label, can never increase as we gather more information.

notion of submodularity to the adaptive setting. As a discrete analog of convexity, in many ways (adaptive) submodularity is a minimal assumption needed to ensure (approximate) tractability. To define adaptive submodularity, we need to extend the definition of the *marginal benefit* Δ_f of an item v to be conditioned on *observations*:

Definition 2.3 (Conditional Expected Marginal Benefit)**.** Suppose we have picked tests $\mathcal{A}$ and have observed outcomes $\mathbf{x}_\mathcal{A}$. The conditional expected marginal benefit of a test v, conditioned on having observed $\mathbf{x}_\mathcal{A}$, is defined as

$$\Delta_f(v \mid \mathbf{x}_\mathcal{A}) = \sum_{x_v} \mathbb{P}\left[X_v = x_v \mid \mathbf{x}_\mathcal{A}\right] \left[f(\mathbf{x}_{\mathcal{A}+v}) - f(\mathbf{x}_\mathcal{A})\right], \tag{2.3.2}$$

where $\mathcal{A} + v$ is the short-hand notation for the $\mathcal{A} \cup \{v\}$.

We further introduce the notion of *subrealization*: We call $\mathbf{x}_\mathcal{A}$ a *subrealization* of $\mathbf{x}_\mathcal{B}$, denoted by $\mathbf{x}_\mathcal{A} \preceq \mathbf{x}_\mathcal{B}$ iff $\mathbf{x}_\mathcal{B}$ contains all observations of $\mathbf{x}_\mathcal{A}$, and possibly some more. The definition of adaptive submodularity is given as follows.

Definition 2.4 (Adaptive Submodularity)**.** A set function $f : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$ is *adaptive submodular* with respect to distribution $\mathbb{P}\left[\mathbf{X}_\mathcal{V}\right]$, if for all subrealizations $\mathbf{x}_\mathcal{A} \preceq \mathbf{x}_\mathcal{B}$ and $v \notin \mathcal{B}$ it holds that

$$\Delta_f(v \mid \mathbf{x}_\mathcal{A}) \geq \Delta_f(v \mid \mathbf{x}_\mathcal{B}).$$

The definition of adaptive monotonicity is given as

Definition 2.5 (Adaptive Monotonicity)**.** A set function $f : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$ is *adaptive monotone* with respect to distribution $\mathbb{P}\left[\mathbf{X}_\mathcal{V}\right]$, if for all subrealizations $\mathbf{x}_\mathcal{A} \preceq \mathbf{x}_\mathcal{B}$ and $v \notin \mathcal{B}$ it holds that $\Delta_f(v \mid \mathbf{x}_\mathcal{A}) \geq 0$.

As illustrated in Figure 2.2, adaptive submodularity is defined with respect to *policies* (as opposed to *sets* in the classical setting) that take observations into account. In

particular, we require that the *expected marginal benefit* associated with any particular querying action never increases as we make more and more observations. One way to see how adaptive submodularity generalizes the classical concept is by interpreting sets as extremely simple policies that ignore observations (and thus don't branch). The same goes for adaptive monotonicity.

2.3.3 The Adaptive Submodular Optimization Framework

A natural approach for adaptive information acquisition is to use a *greedy policy*. Let us denote the adaptive greedy policy as π^g. At each iteration, π^g aims to myopically increase the *expected reward per unit cost*, given its current observations. Formally, we define the expected increase in objective value as follows.

Therefore, if we have observed $\mathbf{x}_\mathcal{A}$, the greedy policy will select the test v maximizing $\Delta_f(v \mid \mathbf{x}_\mathcal{A})/c(v)$. The pseudocode of the adaptive greedy algorithm for the adaptive maximization problem (Problem 2.1.2) is given in Algorithm 1.

Algorithm 1: The adaptive greedy algorithm.

```
1 Input: Budget B; groundset V; Distribution ℙ[X_V]; Reward function f; Cost C;
  begin
2     A ← ∅, x_A ← ∅;
      while C(x_A) ≤ B do
          foreach v ∈ V \ A do
3             Compute Δ_f(v | x_A) = Σ_{x_v} ℙ[X_v = x_v | x_A] [f(x_{A+v}) − f(x_A)];
          end
4         Select v* ∈ arg max_{v∈V\A} Δ_f(v | x_A)/c(v);
5         Set A ← A ∪ {v*}, x_A ← x_A ∪ {x_{v*}};
      end
6     Output: Set A ⊆ V with C(x_A) ≤ B
  end
```

To adapt Algorithm 1 to the adaptive stochastic min-cost cover problem (Problem 2.1.3), we simply modify the stopping condition (i.e., the precondition for the while loop in line 3) to be $f(\mathbf{x}_\mathcal{A}) \geq Q$, such that we keep selecting items as prescribed by π^g until achieving the quota Q on objective value.

Golovin and Krause [GK11b] show that if the reward function f is adaptive submodular and adaptive monotone with respect to the given prior distribution $\mathbb{P}[\mathbf{X}_\mathcal{V}]$, then for both the adaptive submodular maximization and the min-cost coverage problems, the adaptive greedy policy π^g (Algorithm 1) is guaranteed to be competitive with the *optimal policy*.

Theorem 2.6 (Adaptive Submodular Maximization under Cardinality Constraint [GK11b])**.** *Let f be an adaptive monotone and adaptive submodular function with respect to the distribution $\mathbb{P}[\mathbf{X}_\mathcal{V}]$. Suppose we run π^g with budget k. Then for any policies π^* under the same budget,*

$$F(\pi^g) > (1 - 1/e)\, F(\pi^*),$$

where recall $F(\pi) := \mathbb{E}_{\mathbf{x}_\mathcal{V}}[f(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V}))]$ is the expected reward of π.

Theorem 2.7 (Adaptive Submodular Min-cost Cover [GK11b])**.** *Suppose that f is adaptive monotone and adaptive submodular with respect to the distribution $\mathbb{P}[\mathbf{X}_\mathcal{V}]$, and there exists Q such that $f(\mathbf{x}_\mathcal{V}) = Q$ for all $\mathbf{x}_\mathcal{V}$. Let η be any value such that $f(\mathbf{x}_\mathcal{A}) > Q - \eta$ implies $f(\mathbf{x}_\mathcal{A}) = Q$ for all $\mathbf{x}_\mathcal{A}$. Let $\delta = \min_{\mathbf{x}_\mathcal{V}} \mathbb{P}[\mathbf{x}_\mathcal{V}]$ be the minimum probability of any realization. Let π^* be an optimal policy minimizing the expected cost of items selected to guarantee every realization is covered. Then, the expected cost of the greedy policy*

$$\text{cost}_{avg}(\pi^g) \leq \text{cost}_{avg}(\pi^*) \left(\log \frac{Q}{\delta\eta} + 1 \right).$$

Theorem 2.6 and Theorem 2.7 strictly generalize the result of Nemhauser, Wolsey, and Fisher [NWF78] for submodular set functions. These results are very promising, in the sense that they allow characterizing a particular class of optimization problems under partial observability – generally considered notoriously hard – in which efficient algorithms achieve provable guarantees. Due to this fact, adaptive submodularity has been exploited to analyze certain greedy adaptive information acquisition algorithms. For example, Golovin and Krause [GK11b] show that for active learning, the reduction in version space probability mass is adaptive submodular, and prove that the adaptive greedy algorithm is a near-optimal querying policy, recovering and generalizing results of [KPB99a; Das05b] in the learning theory literature. Furthermore, one can utilize an *accelerated adaptive greedy* algorithm applicable to all adaptive submodular functions, which directly uses the diminishing returns property to skip reevaluation of tests, often leading to orders of magnitudes improvement in performance in practice.

Despite these initial, encouraging results, closed-loop submodular optimization remains far less well explored in comparison with the open-loop setting. In many practical scenarios, one can not immediately adopt the adaptive submodular optimization framework due to a number of modeling and computational requirements, such as a known object function which is adaptive submodular, a known prior over a small set of discrete states which can be explicitly enumerated, its limited capability of dealing with noisy test outcomes, etc. We will look into some of these issues, and provide solutions in Part II of this dissertation.

2.3.4 Other Greedy Frameworks for Adaptive Optimization

The Adaptive Worst-case Greedy Algorithm Adaptive stochastic minimum cost coverage (Problem 2.1.3) is related to the (Noisy) Interactive Submodular Set Cover problem [GB10; HY15], which considers the *worst-case* setting (i.e., there is no distribution over states/realizations; instead, states are realized in an adversarial manner). Under such setting, Guillory and Bilmes [GB10] consider a *worst-case greedy* algorithm, which at each step selects the test with maximal worst-case benefit-cost ratio, and prove a logarithmic approximation guarantee for the worst-case cost of such greedy algorithm. Their results rely on the notion of *pointwise submodularity*: a function $f : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$ is pointwise submodular if $f(\mathcal{A}, \mathbf{x}_{\mathcal{V}})$ is submodular in $\mathcal{A}$ for any fixed realization $\mathbf{x}_{\mathcal{V}}$. It is worth pointing out that neither adaptive submodularity nor pointwise submodularity is a strict generalization of the other. While pointwise submodularity is, in general, easier to work with (as one can directly use all standard tools for the analysis of submodular set functions), adaptive submodularity allows deriving bounds on both the average and worst-case policy cost by utilizing the prior $\mathbb{P}[\mathbf{X}_{\mathcal{V}}]$.

The Adaptive Dual Greedy Algorithm The adaptive greedy policy of Algorithm 1 generalizes the greedy algorithm for the (open-loop) subset selection problems [Wol82] (e.g., the classical submodular maximization problem, or the submodular set cover problem). Another well know greedy heuristic under the open-loop setting is the dual greedy algorithm by Fujito [Fuj00], which is based on the dual of Wolsey's LP formulation for submodular set cover (SSC) [Wol82]. The dual greedy algorithm works by setting dual variables greedily to make the dual constraints tight. Since each of the dual constraints corresponds to some test $v \in \mathcal{V}$ in the primal problem, running

the dual greedy algorithms determines an ordering of tests, based on which the SSC solution is constructed. Based on this work, Deshpande, Hellerstein, and Kletenik [DHK14] propose an adaptive extension of the dual greedy algorithm, namely *Adaptive Due Greedy*, and prove an α approximation guarantee for the expected cost. Here, α depends on the cover constructed by the algorithm (e.g., $\alpha = 3$ for the stochastic version of the min-Knapsack problem).

Randomized Greedy Algorithms We have seen that a simple, deterministic greedy policy is useful for maximizing adaptive *monotone* submodular functions. However, adaptive monotonicity may not always hold. For example, in some applications, it may be more natural to formalize Problems 2.1.2 and 2.1.3 as a *scalarized (adaptive maximization)* problem [BV04], e.g., by maximizing the difference between reward and cost, which could lead to non-monotone objective functions.

Unfortunately, the deterministic, discrete greedy algorithm can not handle non-monotone objectives. Under the open-loop setting, Buchbinder et al. [Buc+12] introduce a (randomized) *Double Greedy* algorithm, which achieves a $1/2$ approximation ratio for unconstrained submodular maximization. It maintains two dynamic sets, and in every step, both sets agree (through randomization) whether to include (or exclude) an element in the solution, up to the point where all elements are examined and both sets are identical. Intuitively, randomization is helpful in a sense that it "smooths out" the decisions by not committing to any decisions which are only marginally better.

For the constrained problems, Buchbinder et al. [Buc+14] propose a simple randomized greedy algorithm, which retains the same tight guarantee of $1 - 1/e$ for monotone objectives, while giving an approximation of $1/e$ for general non-monotone objectives. These results are recently generalized to the adaptive setting [GKK15].

Other Greedy Frameworks Analogous to the adaptive submodularity framework for adaptive stochastic submodular optimization, Chan and Farias [CF09] consider a different class of adaptive optimization problems, which is called the *Stochastic Depletion* problems, and show that a greedy policy achieves a $1/2$ approximation of the optimal adaptive policy. Interestingly, one can reduce such setting to capture the stochastic constrained submodular maximization problems, under simple constraints such as cardinality or partition matroid constraints [Cal+07]. However, the general class of the problems is motivated by applications from stochastic optimal control, such as dynamic

product line design (where a firm must dynamically adjust the assortment of products it offers for sale) and stochastic broadcast scheduling (where one must decide which page to broadcast to which group of users). Hence the general line of analysis for these problems are not designed for solving the adaptive information acquisition problems considered in this dissertation.

2.4 Non-myopic Approaches and Probabilistic Planning

Besides those greedy, myopic policies which could be tailored for (closed-loop) submodular optimization, there are also non-greedy techniques, which aim at finding the *optimal* solution. In stochastic optimal control, the decision-theoretic value of information is known as reward functions in *Partially Observable Markov Decision Process* (POMDPs) [SS73], a general framework that captures many adaptive optimization problems under partial observability. In principle, algorithms for planning in POMDPs can be employed for optimizing value of information. Unfortunately, solving POMDPs is PSPACE-hard [PT87], and hence many heuristic search methods without any guarantees have been widely applied [PGT06b; Ros+08].

It is known that the complexity of planning in POMDPs grows exponentially in the cardinality of the state space. For the problems that we consider in this dissertation, the state space grows exponentially with the number of variables (i.e., tests), hence resulting in computational complexity which is doubly exponential in the number of variables. In practice, the idea of using samples to speed up planning has been explored, e.g., POMCP [SV10] which is based on Monte Carlo tree search that samples from states and action histories, and DESPOT [Som+13] which samples scenarios for evaluation of all policies at each iteration. However, these approaches require significant computational resources to generate a finite set of policy to be evaluated. Consequently, they are limited to short planning horizons and relatively small state and action spaces. In contrast, this dissertation aims to design algorithms which could scale to even larger problems. While these POMDP solvers will often use samples to look ahead in the search tree, our results – as we will see in Part II– implies that we do not need to do that, because a simple greedy search (based on some surrogate objective) is already near-optimal.

2.5 Background on Active Learning

The problem of optimal information acquisition has received a lot of attention in the machine learning community in terms of *active learning* (AL). It has been studied in the context of *active instance labeling* or *active feature evaluation*, depending on the objectives of learning: the former is to output a hypothesis by querying data instances (e.g., AL for classification [TK01a]); the latter is to output the value of a instance by actively querying its features (e.g., decision tree learning, or learning with attribute costs [Mit97; KKM05]). Despite the different semantics, structurally the problems are the same, and the goal is to learn a model with small error probability, by using as few labeled examples as possible.

There has been a vast body of work on active learning in the literature, both in theory and applications (for a comprehensive survey see Settles [Set12]). Theses approaches usually differ in the goals and assumptions. In this section, We outline a few variations of active learning problem and highlight some of those works relevant to this dissertation.

2.5.1 Variants of Active Learning Settings

Pool-based AL vs. stream-based AL Active learning algorithms are generally analyzed under two settings: the *pool-based setting* and the *stream-based setting*. In the pool-based setting, the entire set of unlabeled data is available for querying, and the learner can evaluate and rank the entire pool before selecting the best query (examples include text classification [TK01b], image [TC01] and video retrieval [Hau+06], speech recognition [THTS05], and medical diagnosis [Liu04], etc). In the stream-based setting, unlabeled data points arrive in a stream, and the learner must decide whether to ask for their labels or discard the data points, (example applications include spam filtering [Chu+11], streaming document classification [BBB13], and video surveillance [Loy+12], etc.). The stream-based setting is appropriate when it is not possible to postpone selection of data points, for instance, due to storage or timing constraints. A special case of the stream-based setting is called the *secretary setting* [Fer89; BHZ10], in which one must decide immediately whether or not to select a data point at each arrival. What lies in between the secretary and pool-based settings is the case where we can postpone deciding whether or not to query a data point by keeping it in a limited amount of

memory, and at any time query the labels of the stored data points. Algorithms falling into this category are in general also considered as stream-based algorithms.

The adaptive information acquisition problems we study in most chapters are most closely related to the pool-based setting with a Bayesian prior on a (finite) set of hypotheses. For example, similar to the MIS criterion we consider in §3, MacKay [Mac92a] and Cohn [Coh94] propose to pick data points that maximize expected information gain, either in terms of entropy or variance of the estimates. Different from their results which focus on evaluating different selection criteria, we consider the theoretical aspect, where we prove near-optimal guarantees for the adaptive optimization problems.

Noise-free vs. noisy labels Some theoretical results in active classification assume that the observations (i.e., labels of examples) are noise-free. In this case, there is some hypothesis achieving zero error on the data set, and we can use any informative querying strategy to guide the learning process, without the need to worry about the distribution it induces: any inconsistent hypothesis can be eliminated based on a single query, regardless of which distribution this query comes from. Such noise-free setting is also referred to as *realizable* setting.

In contrast, a more practical setting is with noisy tests, which is also known as the *non-realizable* setting, where the optimal hypothesis may not be in the hypothesis class. Even worse, a hypothesis that performs badly on the queried set may well be the optimal hypothesis with respect to the input distribution. This is because of the *sampling bias*: those actively selected data points are no longer *independent and identically distributed* (i.i.d.), and thus the results obtained in the passive setting no longer apply.

Depending on how noise is generated, active learning has been considered under the *non-persistent* noise setting (where the outcome of each test is corrupted independently at random), as well as the *persistent* noise setting (where repeating a test is impossible, or will produce identical observations). The non-persistent noise setting can essentially be reduced to the noise-free setting, simply by repeating each test a sufficient number of times and using the majority of the answers as a proxy for the true outcome [Kää06; Now08] (e.g., if tests have binary outcomes with noise rate α, then by Chernoff's bound [Che81], one can repeat each test $O\left(\frac{\log 1/\delta}{(1/2-\alpha)^2}\right)$ times to obtain the true outcome with probability $1-\delta$). In contrast, the persistent noise setting deals with more stringent, yet practically important noise scenarios: For example, experiments might be systematically biased due to environmental conditions; experts providing labels could make consistent

mistakes. Such noisy setting is often less straightforward to analyze. If we make no assumption on the noise or the error rate of the optimal hypothesis, then the noisy setting is also known as the *agnostic setting* [BBL06; Han07a; Han07b; DHM07; ZC14]. Beside the fully agnostic setting, there are also many other models which require moderate assumptions on the noise, e.g., low noise around the decision boundary [Tsy04; BBZ07; HY14].

Sequential vs. batch-mode. Under the sequential setting, an active learning algorithm chooses labeled examples one by one, each based on the labels of previously selected examples. Instead, batch-mode active learning algorithms [Hoi+06a; GB11a] select a set of examples to be labeled simultaneously. The motivation behind batch active learning is that in some cases it is more cost-effective to request labels in large batches, rather than one-at-a-time. This is true when the cost of labels is sublinear. For instance, if acquiring a label requires running a time-consuming laboratory experiment, or if there is a warm-up time for the equipment, it may be significantly cheaper to run a set of experiments at once. Such setting is also referred to as Buy-in-bulk learning [YC13]. Batch-mode learning algorithms can be in full batch (i.e., open-loop, non-adaptive selection) or partial batches (i.e., closed-loop selection but with multiple queries at each iteration). We investigate both sequential and batch-mode settings in this dissertation.

Active learning vs. active search. As described earlier in this section, classical active learning aims to learn a low error model (e.g., a function for classification or regression) using as few queries as possible. This can be interpreted as active learning at *training time*. In contrast, for some other practical applications, we may not seek to learn the model accurately, but only to find the examples with the maximal *cumulative value*. For example, in recommender systems, we may only care about identifying the *positive examples* in the class of our interest with a minimal number of queries. Such problems are known as *active search* [Gar+12; WGS13], or *adaptive valuable items discovery* [Van+15]. One can also interpret these problems as active learning at *run time*, as it requires to adaptively learn the value functions while picking more examples. Also, it naturally relates to *Multi-armed Bandit*, a class of sequential decision problems characterizing the exploration-exploitation dilemma [CL06]. We will look into this setting in §5, and demonstrate an active search strategy for the object detection application.

Beyond active i.i.d. learning. Most statistical active learning algorithms are developed for i.i.d. supervised learning, where both training and testing data are drawn independently from the same data distribution. For many structured prediction problems, however, the i.i.d. assumption does not hold, as the (structured) output can depend on previous predictions. While it is possible to reduce a structure prediction problem to i.i.d. setting (e.g., by creating a joint feature map of the input and output [Joa+09]), it is often more desirable to view structured prediction as multiple dependent classification tasks, in which each label of the structured output as predicted by a classifier. Techniques for solving such problems are described as imitation learning, or *Learning from demonstration* (LfD) [CT14].

In comparison with classical active i.i.d. learning setting, the class of queries involved in the active LfD problems is much richer: Roughly speaking, there are three types of queries [CT12] of mixed-granularity in a LfD interaction: *label queries* (as in active instance labeling, e.g., "what is the label of this data point?"), *demonstration queries* (as in active class selection [Lom+07], e.g., "can you show me an example of class A?"; or in robotics, "how should I do this given such condition?"), and *feature queries* (as in active feature evaluation [DSM09], e.g., "is this feature important for the target concept being learned?"). There has been some preliminary work evaluating active learning heuristics for LfD problems [CT14]. However, despite its practical significance, little has been known about the theoretical performance for such interactive systems.

2.5.2 Statistical Complexity and Computational Complexity

Statistical Complexity: When does active learning help? In statistical learning theory, the theoretical aim for active learning is to build a thorough understanding of its *statistical complexity* under the *probably approximately correct* (PAC) learning framework, i.e., the number n of examples needed to output a hypothesis that will have expected error at most ϵ with probability at least $1 - \delta$, for some fixed $\epsilon, \delta > 0$. Such complexity measure is known as the *label complexity*, analogous to the term *sample complexity* in the passive model. Due to sampling bias, active learning may require more samples than passive learning to achieve the same generalization error. Therefore, when applying active learning algorithms, we want to make sure that active learning, if not helpful, does not perform worse than passive learning algorithms.

The label complexity of active learning depends on the input distribution and the class

of hypotheses. Intuitively, active learning can be useful if there are always some data points that are considerably more "informative" than others (e.g., when complexity of target function is localized [CWN05]). One can come up with examples where active learning does not help at all, even under the noise-free setting (e.g., see Dasgupta [Das05a]). If labels are noisy with noise rate η, then there is a lower bound of $\Omega\left(\frac{\eta^2}{\epsilon^2}\right)$ on the label complexity of any active learner, i.e., we cannot hope to achieve speedups with when the noise rate is high [Kää06].

Disagreement-based and margin-based active learning. Many active algorithms have been proposed in the past decade, with good statistical properties under various modeling assumptions (see Balcan and Urner [BU15] for a recent survey). In particular, under the stream-based setting, it has been shown that active learning can effectively reduce label complexity, even in the agnostic setting [DMH07; BBL06; Han+11]. This class of algorithms is referred to as *disagreement-based active learning* (DBAL), as the learner queries every example that it is somewhat unsure about (e.g. when there are conflicting hypotheses that disagree with the label of the current example). Such querying strategies also work under the pool-based setting, but are rather conservative compared with most pool-based active leaning approaches which seek out data points that are maximally informative. As a result, disagreement-based active learning often exhibits weaker performance regarding sample complexity.

Under the pool-based setting, often more aggressive approaches are desired. It is worth mentioning that the pool-based algorithms are guaranteed to perform *no worse than* passive learning in terms of sample complexity. The idea is to first draw a pool of n *unlabeled* examples i.i.d. as the initial set. Instead of obtaining all labels, labels are adaptively requested until the labels of all unlabeled data points in the pool are implied by the obtained labels (note that one may have to request all n labels in the worst case). Once the learning process is done, we have obtained n *labeled* points (drawn i.i.d.), and hence classical PAC bounds still apply. An important line of work under the pool-based setting is *margin-based active learning* (MBAL) [BL13a], where learner queries only the data points that are within some margin of the decision boundary. While MBAL achieves a better label complexity than the disagreement-based approaches, these algorithms are often restricted to limited hypothesis classes (e.g., linear separators as in [BL13a; GSSS13]) and restricted noise settings (e.g., [Tsy04; HY14]). A recent work of Zhang and Chaudhuri [ZC14] generalizes MBAL to more general hypothesis spaces.

However their results are not computationally efficient.

Computational complexity. In statistical active learning, algorithms are studied mainly concerning their statistical complexity, disregarding their *computational complexity* [BBL06; ZC14]. An alternative line of work, on the other hand, considers the more aggressive, yet practically-attractive active learning approaches, and focuses on the *optimality* of the algorithm. While the optimal solution is intractable in general [Cha+07], it is of great interest to devise efficient algorithms that are competitive with the optimal algorithm. One famous example of such algorithms is *generalized binary search (GBS)*, or the splitting algorithm, which achieves an $O(\log|\mathcal{H}|)$ approximation [Fre+97; KPB99a; Das05b; Now08] for the expected number of queries in the realizable setting. Here, $|\mathcal{H}|$ denotes the size of the hypothesis class. Such results can also be extended to the noisy setting, where labels of data points are corrupted with independent noise [KK07; Now09]. However, these theoretical analyses rely on the assumption that the labels of data points are *conditionally independent* given any hypothesis, and are not readily extendable to handle more complex dependencies in real-world applications (e.g., in medical diagnosis, conditioned on a diagnosis, the outcomes of medical tests may not be independent). In §3 and §4 of Part II, we will look into different variants of the aggressive active learning problems, and provide near-optimal theoretical guarantees under more relaxed modeling assumptions.

2.6 Related Work in Other Areas

As introduced at the beginning of §1, optimal information acquisition plays a key role in artificial intelligence and sees wide applications in many domains. To get some flavor of the related areas, recall the (partial) lists provided in Fig. 1.1. In this section, we highlight a few areas that are particularly relevant to the theoretical part of this dissertation.

Information theory. Information theory provides powerful tools for quantifying uncertainty in challenging computational problems in data analysis. Naturally, many information-theoretic criteria have been used as design criteria in a variety of fields and applications. The MIS criterion [Lin56] we investigate in this dissertation is perhaps the

most widely used approach in Bayesian active learning and sequential experimental design. The performance bound we obtain in §3 depends on a channel separability condition, which is natural information-theoretic measure that quantifies the severity of noise.

Besides Shannon entropy and mutual information, other information-theoretic criteria have also been studied for adaptive information acquisition, in active learning, sequential hypothesis testing, and other statistical domains. For example, Naghshvar, Javidi, and Chaudhuri [NJC12] study the active sequential hypothesis testing problem, in which they query the label of a sample which maximizes the Extrinsic Jensen-Shannon divergence at each step (motivated by a connection between Bayesian active learning and active hypothesis testing). Meanwhile, the adaptive information acquisition problem also arise in information theory, for example, the generalized binary search algorithm, and its noisy variants, can be viewed as a generalization of Shannon-Fano coding [GS88], channel coding with noiseless feedback [Hor63].

Operations research. Submodular functions are a key concept in operations research and combinatorial optimization. When tests have binary outcomes, the adaptive (submodular) minimum cost cover problem is relevant to *Stochastic Boolean Function Evaluation* (SBFE). In operations research, the SBFE problem is also known as sequential testing of Boolean functions [Ünl04]. In SBFE, one is given a Boolean function $f : \mathbb{B}^n \to \mathbb{B}$, and the task is to evaluate the value of an unknown input $\mathbf{u} \in \mathbb{B}^n$ drawn randomly from a product distribution, by querying the least number of bits u_i. Deshpande, Hellerstein, and Kletenik [DHK14] propose a greedy approach for such problem, and prove it to be near-optimal. However, their analysis only applies to the noise-free case, and it is unclear how these results can be generalized to the noisy setting.

Adaptive sensing and adaptivity gap A natural application domain for information acquisition is sensor networks, where the goal is to deploy a collection of sensors to monitor some spatial phenomenon (e.g., for estimation Krause et al. [Kra+06], or detection [WFIW07]). Krause et al. [Kra+06] show how one can formulate the sensing problem as a submodular optimization problem and provide a near-optimal solution. Moving on to the adaptive setting, a key question is how much better an adaptive policy can perform, when compared to non-adaptive policies that pick the best fixed set of tests a priori. This is quantified by the *adaptivity gap*, which is the worst-case ratio,

between the performance of the optimal adaptive policy and the optimal non-adaptive solution. Asadpour and Nazerzadeh [AN15] show that for the adaptive stochastic (submodular) maximization problem (Problem 2.1.2), the expected value of the optimal non-adaptive policy is at most a constant factor $1 - 1/e$ worse than the expected value of the optimal adaptive policy. Their analysis relies on the assumption that (1) the objective function is submodular, (2) tests have unit cost and (3) test outcomes are independent. In §6 we investigate the adaptivity gap under a more general setting where the independence assumption does not hold, and show how one can utilize such result to exploit information parallelism in adaptive optimization.

Part II

Optimal Value of Information

3

Sequential Information Maximization

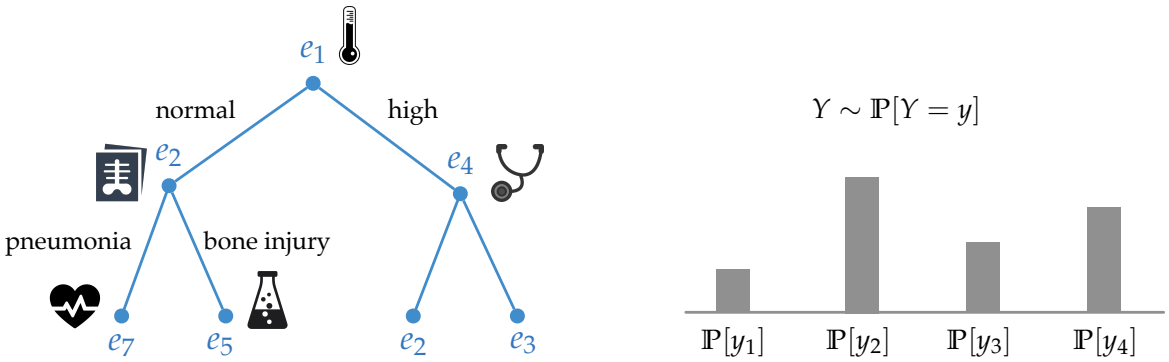
In this chapter, we consider a basic variant of the adaptive information acquisition problem, where the goal is to learn the value of some target random variable through a sequence of *conditional independent*, possibly *noisy* tests. Here, the value of information is defined in terms of the informativeness of the tests performed, measured by Shannon's mutual information. We focus on the adaptive *maximization* variant of the problem (Problem 2.1.2), and hence the name *sequential information maximization*. In particular, we investigate the widely-used adaptive greedy policy, which at each step picks the test that provides the maximal reduction in the Shannon entropy of the target variable. This is the method of choice in numerous applications, such as Bayesian experimental design, automated diagnosis, and active learning, etc. Despite the importance and widespread use, little is known about its theoretical properties, particularly under noisy observations. We provide the first rigorous analysis of such policy that holds under the persistent noise setting (i.e., if repeating a test is impossible or provides no gain).

Organization of this chapter. We start by introducing the basic model in §3.1, and formally state the sequential information maximization problem. In §3.2, we introduce the *channel separability condition*, a natural information-theoretic measure that characterizes the level of noise in the system. Based on such measure, we derive a lower bound on the utility achieved by the greedy policy in §3.3.1, and further give a proof sketch of

the key result in §3.3.2. In §3.4, we provide an example to show that this measure is important in the bound. We conclude this chapter in §3.5.

3.1 Problem Statement

The basic model. Suppose we are given a hidden target random variable Y that ranges among a set $\mathcal{Y} = \{y_1, \cdots, y_n\}$ with some known distribution $Y \sim \mathbb{P}[Y = y]$. The goal is to learn the value of Y by "probing" it with a subset of tests $\mathcal{A} \subseteq \mathcal{V} = \{v_1, \ldots, v_t\}$. Recall that in §2.1.1, we assume that each test $v \in \mathcal{V}$ is associated with some *observable* random variable $X_v \in \mathcal{X}$. In our problem, we can think of the value of Y as representing a true "hypothesis" among a set of possible n hypotheses, and each of the X_v's a "feature" of the target hypothesis which is statistically dependent on Y. In the medical diagnosis example (see Fig. 3.1), Y represents the physical condition of a patient (e.g., possible disease), and $X_1, \ldots, X_t$ represent the patient's symptoms or the outcomes of medical tests. Assume that each test has unit cost. Our goal is to adaptively choose a sequence of k' tests that are maximally informative about Y.

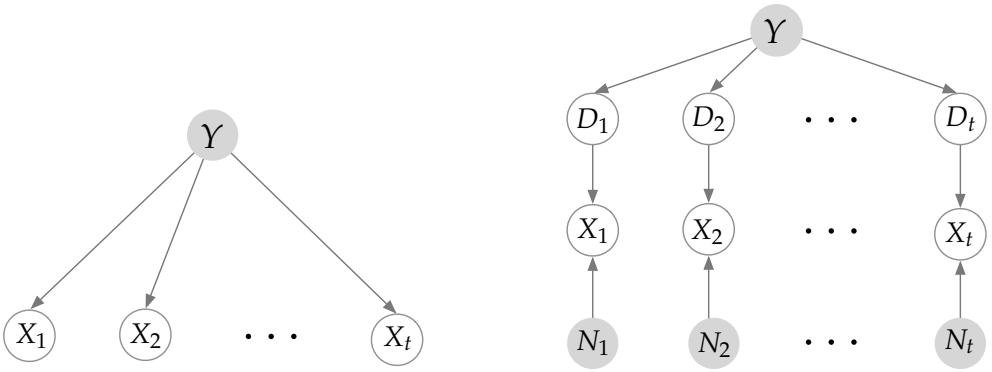


(a) A sequential policy for medical diagnosis

(b) Prior on the target hypothesis

Figure 3.1: Sequential information maximization for medical diagnosis.

We adopt the common assumption in Bayesian experimental design that the joint probability distribution $P(Y, X_1, \ldots, X_t)$ is known and that we can perform efficient inference (i.e., can compute marginal and conditional distributions). In particular, we consider the Naïve Bayes model, i.e., we assume that X_i's are *conditionally independent* given Y (see Fig. 3.2a). Equivalently, we assume that the each test X_i depends on the hidden variable Y and another latent variable N_v, in the following way: First, Y goes



(a) The Naïve Bayes model. (b) An equivalent representation

Figure 3.2: A Bayes net representation of the probabilistic model.

through a deterministic mapping $D_v := D_v(Y)$, i.e, each D_v is a function of Y. The output of D_v will then be perturbed by N_v, and produce the test outcome X_v (see Fig. 3.2b). Hence X_v is a deterministic function of the noise N_v and D_v. Here, one can interpret D_v as the true outcome of test X_v, while the random variable N_v encodes the associated *noise*. Importantly, note that in this construction we assume N_v's to be mutually independent.

Example 3.1 (Generalized Binary Search)**.** *An example for our setting is the generalized binary search (GBS) problem, where Y represents a randomly chosen hypothesis and each X_v is a binary random variable, representing the binary label of example v under hypothesis Y. In the noise-free setting, $X_v = D_v$ is a deterministic function of Y. In the noisy setting, X_v results from flipping the (deterministic) outcome of D_v with probability ϵ and the flipping events of the tests are independent. In other words, we can write $X_v = D_v \oplus N_v$, where D_v is the true label and a deterministic function of Y, N_v is a binary random variable with $Pr(N_v = 1) = \epsilon$, and $\oplus$ denotes the addition in $\mathbb{F}_2 = \{0, 1\}$ (i.e., the XOR operation).*

To show that the graphical model as illustrated in Fig. 3.2b is indeed an equivalent representation of the Naïve Bayes model in Fig. 3.2a, we can consider the following transformation. For any joint distribution $\mathbb{P}[X_1, \ldots, X_t, Y]$, we can write down its joint distribution table, and introduce N as a uniformly-distributed random variable independent of Y, which indexes the rows of $\mathbb{P}[X_1, \ldots, X_t \mid Y]$ according to their conditional probability. In this way, $\mathbb{P}[X_1, \ldots, X_t \mid Y]$ is deterministic given Y and N. Also, N can be properly quantized to be a discrete random variable. This procedure is

similar to the so-called random representation theorem in Markov Chains [LPW, Ch. 1].

Problem statement We consider policies of fixed length, and denote a policy π of length k as $\pi_{[k]}$. Suppose that the realization of all tests in $\mathcal{V}$ is $\mathbf{x}_{\mathcal{V}}$. Then, upon completion, policy $\pi_{[k]}$ returns a sequence of k test-outcome pairs $\mathcal{S}(\pi_{[k]}, \mathbf{x}_{\mathcal{V}}) = \{(v_{\pi,1}, x_{v_{\pi,1}}), (v_{\pi,2}, x_{v_{\pi,2}}), \cdots, (v_{\pi,k}, x_{v_{\pi,k}})\}$. Note that what $\pi_{[k]}$ returns is random, dependent on the (random) outcomes of the selected tests (as well as the decisions that $\pi_{[k]}$ has made). Once $\mathcal{S}(\pi_{[k]}, \mathbf{x}_{\mathcal{V}})$ is observed, we obtain a new posterior of Y, and hence the associated entropy $\mathbb{H}\left(Y \mid \mathcal{S}(\pi_{[k]}, \mathbf{x}_{\mathcal{V}})\right)$. As mentioned in §2.2.1, we define the entropy of Y given the policy $\pi_{[k]}$ as

$$\mathbb{H}\left(Y \mid \pi_{[k]}\right) \triangleq \mathbb{E}_{\mathbf{x}_{\mathcal{V}}}\left[\mathbb{H}\left(Y \mid \mathcal{S}(\pi_{[k]}, \mathbf{x}_{\mathcal{V}})\right)\right]$$

and the mutual information between π and Y is

$$\mathbb{I}\left(\pi_{[k]}; Y\right) = \mathbb{H}\left(Y\right) - \mathbb{H}\left(Y \mid \pi_{[k]}\right)$$

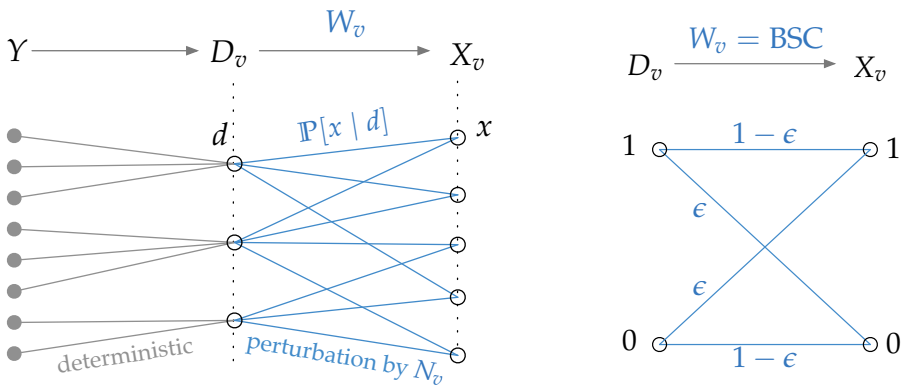
which indicates the expected amount of information that $\pi_{[k]}$ provides about Y upon completion. We then define the optimal policy $\pi_{\mathrm{OPT}[k]}$ to be the policy that achieves the maximal expected mutual information, i.e.,

$$\pi_{\mathrm{OPT}[k]} = \arg\max_{\pi \in \Pi_{[k]}} \mathbb{I}\left(\pi_{[k]}; Y\right) \tag{3.1.1}$$

where $\Pi_{[k]}$ is the set of all policies of length k. Note that computing the optimal policy is intractable in general. A very well known, efficient and intuitive policy is the one that greedily picks the test that reduces the current entropy of Y the most, or equivalently, has the maximum mutual information w.r.t. the current distribution of Y. We denote such *most informative selection policy* of length k by $\pi_{\mathrm{MIS}[k]}$. Then, at round ℓ, $\pi_{\mathrm{MIS}[k]}$ picks a test

$$v^* \in \arg\max_{v \in \mathcal{V}} \mathbb{I}\left(X_v; Y \mid \mathcal{S}(\pi_{\mathrm{MIS}[\ell-1]}, \mathbf{x}_{\mathcal{V}})\right). \tag{3.1.2}$$

In the following sections, we will provide a detailed analysis of the MIS policy, and establish conditions under which MIS achieves near-optimal performance.



(a) Illustration of the noise channel

(b) BSC

Figure 3.3: Illustration of the channel induced by noise. (a) shows the data generation process. In (b) we illustrate the binary symmetric channel for Example 3.1.

3.2 Channel Induced by Noise

Intuitively, the performance of the policy should depend on the amount of uncertainty in the system. In this section, we quantify the noise in the system in terms of a natural channel separability condition.

Recall that for any $e \in \mathcal{V}$, the random variable D_v is a deterministic function of Y. The value of D_v is then perturbed by the noise N_v to generate the test variable X_v. Since the perturbation of N_v is assumed to take place independently of Y, we can characterize such perturbation through a conditional probability distribution $\mathbb{P}[X_v = x \mid D_v = d]$ where $x \in \mathcal{X}$, $d \in \mathcal{D}$, and $\mathcal{D}$, $\mathcal{X}$ are the support of D_v, X_v. We refer to this conditional probability distribution as *the channel induced by the noise* and denote it by W_v (see Figure 3.3).

The test X_v depends on Y only through D_v, i.e., we have the Markov chain $Y \to D_v \to X_v$. As a result,

$$\mathbb{I}(X_v; Y) = \mathbb{I}(X_v; D_v),$$

and the latter is by definition less than the capacity of the channel W_v. We now introduce another parameter for the channel W_v and will later establish its importance for sequential selection.

Definition 3.2 (Separability of a channel)**.** Consider a channel W with associated conditional probability distribution $\{\mathbb{P}[x \mid d]\}_{d \in \mathcal{D}, x \in \mathcal{X}}$. Note that given each $d \in \mathcal{D}$,

$\mathbb{P}[\cdot \mid d]$ is a probability distribution over $\mathcal{X}$. The separability of W, denoted by $S(W)$, is then defined by

$$S(W) = \left(\min_{d,d' \in \mathcal{D}: d \neq d'} \left| \mathbb{P}[\cdot \mid d] - \mathbb{P}\left[\cdot \mid d'\right] \right|_{\mathrm{TV}} \right)^2. \tag{3.2.1}$$

Here, $|\cdot|_{\mathrm{TV}}$ denotes the total variation distance. Also, if $|\mathcal{D}| = 1$ we let $S(W) = 1$. Intuitively, for a channel W and two inputs d, d', the value $|\mathbb{P}[\cdot \mid d] - \mathbb{P}[\cdot \mid d']|_{\mathrm{TV}}$ is an indicator of how much the channel can differentiate between d and d'. E.g., if $|\mathbb{P}[\cdot \mid d] - \mathbb{P}[\cdot \mid d']|_{\mathrm{TV}} = 0$ then $\mathbb{P}[\cdot \mid d] = \mathbb{P}[\cdot \mid d']$, in which case it is impossible to distinguish d from d' given the output of the channel. On the other hand, if $|\mathbb{P}[\cdot \mid d] - \mathbb{P}[\cdot \mid d']|_{\mathrm{TV}} = 1$ then from the output we can for sure exclude either d or d' (i.e., if we know that the input was either d or d', then we can say from the output which one is the input).

As mentioned above, for any $v \in \mathcal{V}$ we have an associated channel W_v which is induced by the noise N_v. We denote by $S_{\min}$ the minimum value of separability over all the channels W_v, i.e.,

$$S_{\min} = \min_{v \in \mathcal{V}} S(W_v).$$

For the noisy GBS example (Example 3.1), it is easy to see that the separability of the binary symmetric channel (see Fig. 3.3b) is $S_{\min} = (1 - 2\epsilon)^2$.

3.3 A Lower Bound on the Utility

Now we are ready to state our main result, which provides the first approximation guarantee on the performance of the MIS policy under the persistent noise setting.

3.3.1 Main Result

Theorem 3.3. *Consider the sequential information maximization problem, where we run the most informative selection policy π_{MIS} till length k'. For any $\delta > 0$ and $k \in \mathbb{N}$, we have*[1]

$$\mathbb{I}\left(\pi_{\mathrm{MIS}[k']}; Y\right) \geq \left(\mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right) - \delta\right)\left(1 - \exp\left(-\frac{k'}{k\gamma \max\{\log n, \log \frac{1}{\delta}\}}\right)\right), \tag{3.3.1}$$

[1] Throughout this dissertation, all the log's are in base 2.

where $n = |\mathcal{Y}|$ is the number of possible values of Y, and γ is a constant that only depends on the noise N, concretely: $\gamma = \frac{7}{S_{\min}}$.

To avoid a lengthy interruption of the exposition, we defer the proof of Theorem 3.3 to §3.3.2. Here, we list a few noteworthy observations. First, suppose that for some fixed $0 < \alpha < 1$ we have that $\delta = \alpha \mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right)$. Thus, δ is expressed as a fraction of the maximum mutual information obtainable by any policy. Then the RHS of Inequality (3.3.1) turns into a multiplicative bound in terms of α:

$$\mathbb{I}\left(\pi_{\mathrm{MIS}[k']}; Y\right) \geq \mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right)(1-\alpha)\left(1 - \exp\left(-\frac{k'}{k\gamma \max\{\log n, \log \frac{1}{\alpha \mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right)}\}}\right)\right).$$

We note that for many reasonable scenarios, $\mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right)$ is typically at least a few bits. Otherwise arguably the information gathering task is ill-posed / infeasible. In this case, if we assume $\mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right) \geq 1$, then we obtain a lower bound where the multiplicative factor only depends on the noise channel:

$$\mathbb{I}\left(\pi_{\mathrm{MIS}[k']}; Y\right) \geq \mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right)(1-\alpha)\left(1 - \exp\left(-\frac{k'}{k\gamma \max\{\log n, \log \frac{1}{\alpha}\}}\right)\right).$$

Another way to interpret the result is to use the fact that $I(\pi_{\mathrm{OPT}[k]}; Y) \leq \log n$. From (3.3.1) we obtain

$$\mathbb{I}\left(\pi_{\mathrm{MIS}[k']}; Y\right) \geq \mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right) - \delta - \log n\left(1 - \exp\left(-\frac{k'}{k\gamma \max\{\log n, \log \frac{1}{\delta}\}}\right)\right),$$

As a consequence, if we choose $k' \geq k\gamma \max\{\log n, \log \frac{1}{\delta}\} \ln(\frac{\log n}{\delta})$, then we have

$$\mathbb{I}\left(\pi_{\mathrm{MIS}[k']}; Y\right) \geq \mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right) - 2\delta.$$

Hence, we can get arbitrarily close – up to δ in absolute terms – to the optimal mutual information achievable within k tests by greedily selecting k' tests, which is a logarithmic factor (in terms of $\log n$ and $\log \frac{1}{\delta}$) of k.

Remark 3.4. A few comments are in order. First, as an example, for the GBS problem in Example 3.1, $\gamma = \frac{7}{(1-2\epsilon)^2}$, and the lower bound we get for the greedy algorithm is $\mathbb{I}\left(\pi_{\mathrm{MIS}[k']}; Y\right) \geq \left(\mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right) - \delta\right)\left(1 - \exp(\frac{k'(1-2\epsilon)^2}{7k \max\{\log n, \log \frac{1}{\delta}\}})\right)$.
Second, let us assume that in expectation, the policy π_{OPT} manages to reduce the uncertainty of Y to δ. Denote the cost of such a policy as $C_{\mathrm{OPT}}(\delta)$. In other words,

we have $I\left(\pi_{\text{OPT}[C_{\text{OPT}}(\delta)]}; Y\right) \geq \mathbb{H}(Y) - \delta$. Further, let $\delta' = 2\delta$. We ask how many tests the greedy policy π_{MIS} needs, call it $C_{\text{MIS}}(\delta')$, so that $I\left(\pi_{\text{MIS}[C_{\text{MIS}}(\delta')]}; Y\right) \geq \mathbb{H}(Y) - \delta'$. From Theorem 3.3, it is not difficult to compute that π_{MIS} needs at most

$$\begin{aligned} C_{\text{MIS}}(\delta') &\leq C_{\text{OPT}}(\delta) \cdot \gamma \cdot \max\{\log n, \log \frac{1}{\delta}\} \cdot \ln(\frac{\mathbb{H}(Y)}{\delta}) \\ &\leq C_{\text{opt}}(\delta) \cdot \gamma \cdot \max\{\log n, \log \frac{1}{\delta}\} \cdot \ln(\frac{\log(n)}{\delta}). \end{aligned}$$

Therefore, for the greedy policy to gain information "close to" $\pi_{\text{OPT}[k]}$, we need to run $\pi_{\text{MIS}[k']}$ for $k' = O\left(k \cdot \frac{\log n}{S_{\min}}\right)$ rounds. For the GBS problem, we have $k' = O\left(k \cdot \frac{\log n}{(1-2\epsilon)^2}\right)$. Thus, when $S_{\min}$ is large (e.g., ϵ is small), then the greedy policy exhibits near-optimal performance.

Third, the $S_{\min}$ involved in our bound is defined over all the possible tests *picked by* π_{MIS} or π_{OPT}. Therefore, if there are some tests which are "purely noisy", i.e., the separability of their associated noise channels have $S(W) = 0$, then clearly both π_{MIS} and π_{OPT} will disregard those tests, and hence their $S(W)$'s don't affect our lower bound.

3.3.2 The Analysis Framework

In this section, we prove Theorem 3.3. A key lemma in proving the theorem is as follows.

Lemma 3.5. *Consider the probabilistic model of Fig. 3.2 with an arbitrary probability distribution $\mathbb{P}[\cdot]$. Also, consider any adaptive policy π which chooses k tests among $\{X_v\}_{v\in\mathcal{V}}$ and gains mutual information $\mathbb{I}(\pi; Y)$. Then, we must have*

$$\max_{v\in\mathcal{V}} \mathbb{I}(X_v; Y) \geq \frac{\mathbb{I}(\pi; Y)}{k\gamma \max\left\{\log n, \log \frac{1}{\mathbb{I}(\pi;Y)}\right\}}. \tag{3.3.2}$$

We relegate the proof of this lemma to the next section. Let us now see how the result of Theorem 3.3 follows from this lemma.

Proof of Theorem 3.3. Now, we show that Eq. (3.3.1) holds for any policy $\pi_{[k]}$ of length k. For simplicity, let

$$\Psi_\ell \triangleq \mathcal{S}(\pi_{\text{MIS}[\ell]}, \mathbf{X}_\mathcal{V})$$

be a random variable representing the first ℓ tests (and their associated outcomes) that have been selected by the greedy policy π_{MIS}, and $\psi_\ell \triangleq \mathcal{S}(\pi_{\text{MIS}[\ell]}, \mathbf{x}_\mathcal{V})$ be a specific realization of Ψ_ℓ. In the decision tree representation of $\pi_{\text{MIS}[\ell]}$, ψ_ℓ represents a path from the root to a node at level ℓ (see Fig. 3.4). Now suppose we have run the greedy policy π_{MIS} till level ℓ, and have observed the realized path ψ_ℓ (thus ψ_ℓ is a sequence of ℓ chosen tests and their observed outcomes). At this point, π_{MIS} picks a new test according to the greedy rule (Eq. (3.1.2)). Therefore, the expected gain of the greedy algorithm at time $\ell + 1$ is $\max_{v \in \mathcal{V}} \mathbb{I}(X_v; Y \mid \psi_\ell)$.

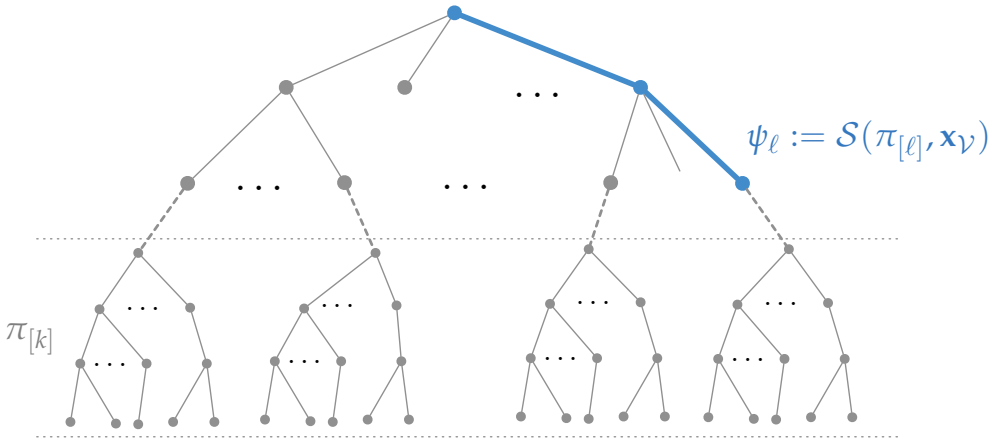


Figure 3.4: The decision tree representation of policies (i) $\pi_{\text{MIS}[\ell]}$ of length ℓ, and (ii) $\pi_{[k]}$ of length k. After π_{MIS} has selected ℓ tests (observed ψ_ℓ), we run policy $\pi_{[k]}$, as if from a fresh start. This is known as the concatenation of the two policies $\pi_{\text{MIS}[\ell]}$ and $\pi_{[k]}$, see [GK11a].

Let us now consider the following thought experiment. Assume the same setting as above (i.e., we have observed ψ_ℓ) and we run the policy $\pi_{[k]}$ of length k as if from a fresh start (Fig. 3.4), i.e., $\pi_{[k]}$ is run by totally neglecting the observation ψ_ℓ. The policy $\pi_{[k]}$ then outputs a realization ψ_π. The expected information we obtain by using the aforementioned version of π (that totally neglects the observation ψ_ℓ) is $\mathbb{H}(Y \mid \psi_\ell) - \mathbb{H}\left(Y \mid \psi_\ell, \pi_{[k]}\right)$ or equivalently $\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right)$. We can now use the result of Lemma 3.5 to relate the gain of the greedy to the gain of $\pi_{[k]}$. An important point to note here is that the result of Lemma 3.5 holds for *any* probability distribution on the Bayesian network of Fig. 3.2. In particular, by conditioning all our distributions

on the observation ψ_ℓ, and by using Lemma 3.5, we obtain

$$\max_{v\in\mathcal{V}} \mathbb{I}\left(X_v; Y \mid \psi_\ell\right) \geq \mathbb{E}_{\psi_\ell}\left[\frac{\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right)}{k\gamma \max\{\log n, \log \frac{1}{\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right)}\}}\right]. \tag{3.3.3}$$

Now, by further averaging over ψ_ℓ, the expected entropy reduction by running $\pi_{[k]}$ after $\pi_{\mathrm{MIS}[\ell]}$ is

$$\begin{aligned}
\mathbb{E}_{\psi_\ell}\left[\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right)\right] &= \mathbb{E}_{\psi_\ell}\left[\mathbb{H}\left(Y \mid \psi_\ell\right) - \mathbb{H}\left(Y \mid \pi_{[k]}, \psi_\ell\right)\right] \\
&= \mathbb{H}\left(Y \mid \pi_{\mathrm{MIS}[\ell]}\right) - \mathbb{H}\left(Y \mid \pi_{[k]}, \pi_{\mathrm{MIS}[\ell]}\right) \\
&\geq \mathbb{H}\left(Y \mid \pi_{\mathrm{MIS}[\ell]}\right) - \mathbb{H}\left(Y \mid \pi_{[k]}\right) \\
&= \mathbb{I}\left(\pi_{[k]}; Y\right) - \mathbb{I}\left(\pi_{\mathrm{MIS}[\ell]}; Y\right).
\end{aligned} \tag{3.3.4}$$

Note here that $\mathbb{I}\left(\pi_{[k]}; Y\right)$ is the total information gain of the policy $\pi_{[k]}$ about Y. Fix $\delta > 0$, and denote $\alpha := k\gamma \max\{\log n, \log \frac{1}{\delta}\}$. We can resume Eq. (3.3.3) as follows

$$\begin{aligned}
\mathbb{E}_{\psi_\ell}\left[\max_{v\in\mathcal{V}} \mathbb{I}\left(X_v; Y \mid \psi_\ell\right)\right] &\overset{\text{(inequality (3.3.3))}}{\geq} \mathbb{E}_{\psi_\ell}\left[\frac{\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right)}{k\gamma \max\{\log n, \log \frac{1}{\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right)}\}}\right] \\
&\geq \mathbb{E}_{\psi_\ell}\left[\frac{\mathbb{I}\left(\pi_{[k]}; Y \mid \psi\right) \cdot \mathbb{1}\left\{\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right) > \delta\right\}}{k\gamma \max\{\log n, \log \frac{1}{\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right)}\}}\right] \\
&\geq \frac{1}{\alpha}\left(\mathbb{E}_{\psi_\ell}\left[\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right) \cdot \mathbb{1}\left\{\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right) > \delta\right\} + \delta\right] - \delta\right) \\
&\geq \frac{1}{\alpha}\left(\mathbb{E}_{\psi_\ell}\left[\mathbb{I}\left(\pi_{[k]}; Y \mid \psi_\ell\right)\right] - \delta\right) \\
&\overset{\text{(inequality (3.3.4))}}{\geq} \frac{1}{\alpha}\left(\mathbb{I}\left(\pi_{[k]}; Y\right) - \mathbb{I}\left(\pi_{\mathrm{MIS}[\ell]}; Y\right) - \delta\right).
\end{aligned}$$

Rearranging the terms, we have

$$\begin{aligned}
\mathbb{I}\left(\pi_{[k]}; Y\right) - \delta - \mathbb{I}\left(\pi_{\mathrm{MIS}[\ell]}; Y\right) &\leq \alpha \mathbb{E}_{\psi_\ell}\left[\max_{v\in\mathcal{V}} \mathbb{I}\left(X_v; Y \mid \psi_\ell\right)\right] \\
&= \alpha\left(\mathbb{H}\left(Y \mid \pi_{\mathrm{MIS}[\ell]}\right) - \mathbb{E}_{\psi_\ell}\left[\min_{v\in\mathcal{V}} \mathbb{H}\left(Y \mid X_v, \psi_\ell\right)\right]\right) \\
&= \alpha\left(\mathbb{I}\left(\pi_{\mathrm{MIS}[\ell+1]}; Y\right) - \mathbb{I}\left(\pi_{\mathrm{MIS}[\ell]}; Y\right)\right).
\end{aligned} \tag{3.3.5}$$

Let $\Delta_\ell := \mathbb{I}\left(\pi_{[k]}; Y\right) - \delta - \mathbb{I}\left(\pi_{\mathrm{MIS}[\ell]}; Y\right)$, so that Inequality (3.3.5) implies $\Delta_\ell \leq$

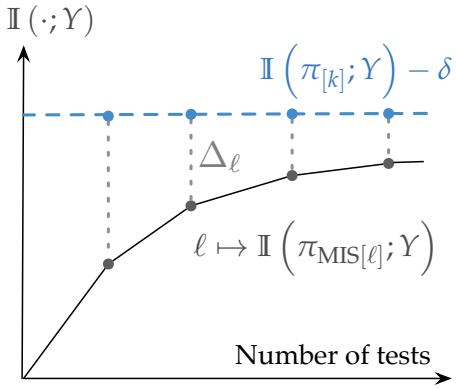


Figure 3.5: Bounding Δ_0 against Δ_ℓ.

$\alpha \cdot (\Delta_\ell - \Delta_{\ell+1})$. From here we get $\Delta_{\ell+1} \leq \left(1-\frac{1}{\alpha}\right)\Delta_\ell$, and hence $\Delta_{k'} \leq \left(1-\frac{1}{\alpha}\right)^{k'}\Delta_0 \leq \exp\left(-\frac{k'}{\alpha}\right)\Delta_0$. See Fig. 3.5 for illustration.

Substituting Δ'_k, Δ_0 with their definitions, we get

$$\mathbb{I}\left(\pi_{[k]};Y\right)-\delta-\mathbb{I}\left(\pi_{\mathrm{MIS}[k']};Y\right) < \exp\left(-\frac{k'}{\alpha}\right)\Delta_0 \leq \exp\left(-\frac{k'}{\alpha}\right)\left(\mathbb{I}\left(\pi_{[k]};Y\right)-\delta\right).$$

This gives us $\mathbb{I}\left(\pi_{\mathrm{MIS}[k']};Y\right) \geq \left(\mathbb{I}\left(\pi_{[k]};Y\right)-\delta\right)\left(1-\exp\left(-\frac{k'}{k\gamma\max\{\log n,\log\frac{1}{\delta}\}}\right)\right)$. □

3.3.3 Proof Sketch of the Key Lemma

We first show a sufficient condition for Lemma 3.5 as follows.

Lemma 3.6. *Fix any $\alpha \in [0,\log n]$. If we assume $\max_{v\in\mathcal{V}}\mathbb{I}(X_v;Y)$ is sufficiently small, i.e.,*

$$\max_{v\in\mathcal{V}}\mathbb{I}(X_v;Y) \leq \frac{\alpha}{k\gamma\max\{\log n,\log\frac{1}{\alpha}\}} \triangleq I_0(\alpha). \tag{3.3.6}$$

then no policy of length k can achieve a mutual information α, i.e., $\forall\pi_{[k]}$, $\mathbb{I}\left(\pi_{[k]};Y\right) < \alpha$.

The sufficiency is immediate: suppose Lemma 3.6 holds. Now, if there exists a policy π of length k with mutual information $\mathbb{I}(\pi;Y)$, then by letting $\alpha = \mathbb{I}\left(\pi_{[k]};Y\right)$, it must hold that $\max_{v\in\mathcal{V}}\mathbb{I}(X_v;Y) > \frac{\mathbb{I}\left(\pi_{[k]};Y\right)}{k\gamma\max\{\log n,\log\frac{1}{\mathbb{I}\left(\pi_{[k]};Y\right)}\}}$, which gives us Lemma 3.5.

So for the rest of the proof, we focus on proving Lemma 3.6. We assume that (3.3.6) holds. We consider a policy[2] π of length k and show that $\mathbb{I}(\pi, Y) < \alpha$. We divide the proof into three steps.

Step 1: Recall that each r.v. D_v, as a deterministic function of Y, takes values over its support $\mathcal{D}_v$ with distribution $\mathbb{P}[D_v = d_v]$ (and these distributions are possibly different for each $e \in \mathcal{V}$). Denote $p_{v,\max} = \max_{d_v \in \mathcal{D}_v} \mathbb{P}[D_v = d_v]$ to be the probability of the most likely outcome for D_v. Let us first see how the value $\mathbb{I}(X_v; Y)$ can be expressed in terms of $p_{v,\max}$ and $S(W_v)$. We first argue that for any $e \in \mathcal{V}$ we have that $\mathbb{I}(X_v; Y) = \mathbb{I}(X_v; D_v)$. This is because D_v is a function of Y, and X_v is a function of noise N (which is independent of Y) and D_v (i.e., $Y \to D_v \to X_v$ forms a Markov chain).

Lemma 3.7. *Fix $\theta \in (0, 1/4]$. If $\mathbb{I}(X_v; D_v) \leq \theta S(W_v)$, then we have $p_{v,\max} \geq (1 + \sqrt{1 - 4\theta})/2$.*

We relegate the proof of this lemma to §A.1.1. By combining Lemma 3.7 with Inequality (3.3.6) and the fact that $\mathbb{I}(X_v; Y) = \mathbb{I}(X_v; D_v)$, we obtain that for any $v \in \mathcal{V}$ (note the fact that $I_0 / S_{\min} \leq \frac{1}{4}$),

$$p_{v,\max} \geq \frac{1}{2}\left(1 + \sqrt{1 - 4\theta}\right) \geq \frac{1}{2}\left(1 + \sqrt{1 - 4\frac{I_0}{S_{\min}}}\right) \triangleq \beta \tag{3.3.7}$$

Step 2: This is the situation at level 0. To investigate how the values $p_{v,\max}$ change as we perform more tests and observe their outcomes, we have to take into account how the noise affects the prior and so on. However, we intend to avoid such an analysis. For this purpose, we first prove that the performance of the system would only become better if we were given full information about the noise N. Formally speaking, as mentioned above, a policy π which has length k can also be thought of as a random object which outputs a set of k test-outcome pairs $\psi_\pi \triangleq \{(v_{\pi,1}, x_{v_{\pi,1}}), \ldots, (v_{\pi,k}, x_{v_{\pi,k}})\}$. Such a policy starts from a root node (with only the knowledge about the probabilistic model of Y and X_v's) and performs its tests sequentially and adaptively according to what it has observed. Now, consider another random variable G defined as follows:

$$G = \{(v_{\pi,1}, d_{v_{\pi,1}}), (v_{\pi,2}, d_{v_{\pi,2}}), \ldots, (v_{\pi,k}, d_{v_{\pi,k}})\}. \tag{3.3.8}$$

One can think of G as an *oracle* sitting beside the system π and observing its actions

[2] To simplify notation, in what follows in this chapter we use π instead of $\pi_{[k]}$.

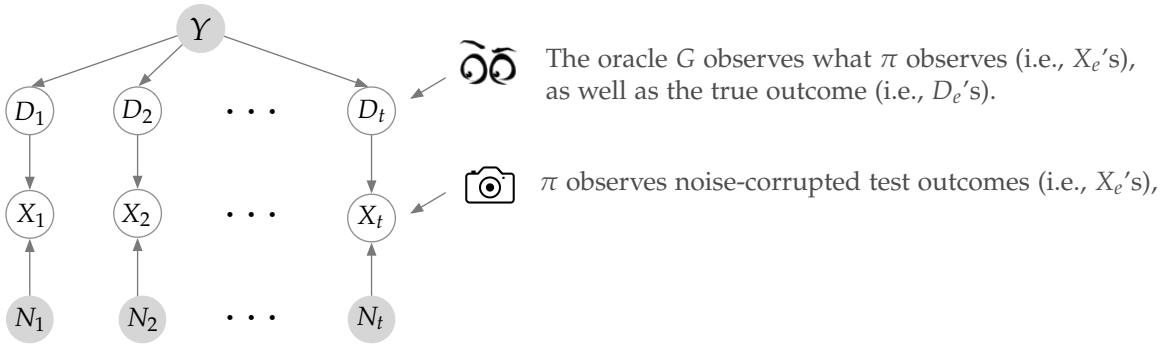


Figure 3.6: Illustration of Step 2: An *oracle* G sitting beside the system π.

(see Fig. 3.6). Furthermore, at each time whatever test e that π picks, G has access to the outcome of D_v, i.e., d_v. Note that G does not know the true value of Y. But we expect that G has a better idea about Y than π has. This is because G knows the deterministic outcomes of the tests that π has picked while π only knows a noisy version of these deterministic outcomes (i.e., what π observes is a noisy version of what G observes). Let N be a random vector concatenating all N_v. Indeed, we can write

$$\begin{aligned}\mathbb{I}(G;Y) &= \mathbb{H}(Y) - \mathbb{H}(Y \mid G) \\ &\overset{(a)}{=} \mathbb{H}(Y) - \mathbb{H}(Y \mid G, N) \\ &\overset{(b)}{\geq} \mathbb{H}(Y) - \mathbb{H}(Y \mid \pi) = \mathbb{I}(\pi;Y),\end{aligned} \tag{3.3.9}$$

where (a) follows from the fact that Y is independent of N, and (b) is because the output of π is a deterministic function of the the noise N and the output of G. The idea now is to analyze G. Note that: (i) Since G has access to the deterministic values d_v of the tests that π picks, its posterior about Y is decoupled from the noise N. (ii) Any upper bound on $\mathbb{I}(G;Y)$ would also be an upper bound on $\mathbb{I}(\pi;Y)$ by (3.3.9).

Step 3: Let us now find an upper bound on $\mathbb{I}(G;Y)$. For this, we start from the root node of π. Recall that in Step 1 we proved that any of the tests X_v satisfies the relation (3.3.7). In other words, at time 0 (before performing any tests by π), if we define for $e \in \mathcal{V}$

$$b_v = \arg\max_{b \in \mathcal{D}} \{\mathbb{P}[D_v = b]\}, \text{ and } \mathcal{Y}_v = \{y \in \mathcal{Y} : D_v(y) = b_v\}, \tag{3.3.10}$$

then by (3.3.7) we have that

$$\mathbb{P}\left[y \in \mathcal{Y}_v^{\complement}\right] = \mathbb{P}[D_v(Y) \neq b_v] = 1 - p_{v,\max} \leq 1 - \beta, \tag{3.3.11}$$

where by $\mathcal{Y}_v^{\complement}$ we mean the complement of the set $\mathcal{Y}_v$. The policy π has length k, i.e., it sequentially and adaptively performs tests which we denote by $v_{\pi,1}, \cdots, v_{\pi,k}$ and the choice of $v_{\pi,i}$ is based on the full observation of the outcomes of $v_{\pi,1}, \cdots, v_{\pi,i-1}$. We now consider the following event

$$\Lambda = \{(D_{v_{\pi,1}} = b_{v_{\pi,1}}) \wedge (D_{v_{\pi,2}} = b_{v_{\pi,2}}) \wedge \cdots \wedge (D_{v_{\pi,k}} = b_{v_{\pi,k}})\}, \tag{3.3.12}$$

I.e., $\mathcal{A}$ is the event that whatever test e that π picks, its deterministic part D_v outputs its most likely outcome b_v. See Fig. 3.7 for illustration.

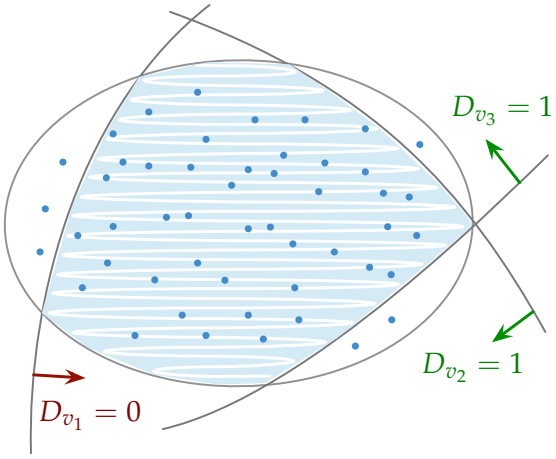


Figure 3.7: Illustration of Step 3: Solid dots represent hypotheses, and the lines represent (binary) tests. Large marginal probabilities ($\mathbb{P}[D_{v_1} = 0]$, $\mathbb{P}[D_{v_2} = 1]$ and $\mathbb{P}[D_{v_3} = 1]$) imply a large joint probability $\mathbb{P}[D_{v_1} = 0, D_{v_2} = 1, D_{v_3} = 1]$ (c.f., Eq. (3.3.14)).

We establish a lower bound on the probability of Λ through the following lemma:

Lemma 3.8. *If for every test $v \in \mathcal{V}$ we have $p_{v,\max} \geq \beta$, then $\mathbb{P}[\Lambda] \geq 1 - k(1-\beta)$.*

We relegate the proof of this lemma to §A.1.2. In other words, the random variable G has observations compatible with the event Λ with probability at least as the lower bound provided in Lemma 3.8. We can now write

$$\begin{aligned} \mathbb{H}(Y \mid G) &\overset{(a)}{=} \mathbb{P}[\Lambda]\,\mathbb{H}(Y \mid G, \Lambda) + (1 - \mathbb{P}[\Lambda])\mathbb{H}\left(Y \mid G, \Lambda^{\complement}\right) \\ &\overset{(b)}{\geq} \mathbb{P}[\Lambda]\,\mathbb{H}(Y \mid G, \Lambda)\,. \end{aligned} \tag{3.3.13}$$

Here, (a) follows from the fact that the event Λ is a function of what G observes. Also, (b) follows from entropy function being positive. It remains to find a lower bound

on $\mathbb{H}(Y \mid G, \Lambda)$. For this, note that if we end up being in the event Λ, then all the hypotheses in the set $\cap_{j=1}^{k} \mathcal{Y}_{v_{\pi,j}}$ would remain compatible with the observations that G has had. Let us assume that G has observed $\{v_{\pi,1}, \ldots, v_{\pi,k}\}$ and the corresponding outcomes $\{D_{v_{\pi,1}} = b_{v_{\pi,1}}, \ldots, D_{v_{\pi,k}} = b_{v_{\pi,k}}\}$ (so that event Λ has taken place). To simplify notation, let us define $U \triangleq \cap_{j=1}^{k} \mathcal{Y}_{v_{\pi,j}}$. By the union bound and Eq. (3.3.7) we have

$$\mathbb{P}[U] = 1 - \mathbb{P}\left[U^{\complement}\right] \geq 1 - \sum_{j=1}^{k} (1 - \mathbb{P}\left[\mathcal{Y}_{v_{\pi,j}}\right]) \geq 1 - k(1-\beta). \tag{3.3.14}$$

Now, the posterior that G has about Y, $\mathbb{P}[Y \mid y \in U]$, would become as follows:

$$\text{If } y \in U, \text{then } \mathbb{P}[y \mid y \in U] = \frac{\mathbb{P}[y]}{\mathbb{P}[U]}, \qquad \text{and} \qquad \text{If } y \notin U, \text{then } \mathbb{P}[y \mid y \in U] = 0.$$

The entropy of the posterior then becomes

$$\begin{aligned}
\mathbb{H}(Y \mid y \in U) &= \sum_{y \in U} \frac{\mathbb{P}[y]}{\mathbb{P}[U]} \log \frac{\mathbb{P}[U]}{\mathbb{P}[y]} \\
&= \frac{1}{\mathbb{P}[U]} \sum_{y \in X} \mathbb{P}[y] \log \frac{1}{\mathbb{P}[y]} + \log \mathbb{P}[U] \\
&\overset{(a)}{\geq} \frac{\mathbb{H}(Y)}{\mathbb{P}[U] - \frac{1-\mathbb{P}[U]}{\mathbb{P}[U]} \log \frac{n}{1-\mathbb{P}[U]} + \log \mathbb{P}[U]} \\
&\overset{(b)}{\geq} \mathbb{H}(Y) - \frac{1-\rho}{\rho} \log \frac{n}{1-\rho} + \log(\rho).
\end{aligned} \tag{3.3.15}$$

Here, step (a) follows Lemma A.2 as stated in §A.1.2. In step (b) we have assumed that $\rho \triangleq 1 - k(1-\beta)$. We thus have from (3.3.14) that $\mathbb{P}[U] \geq \rho$, and step (b) follows from simple calculus. We also note from Lemma 3.8 that $\mathbb{P}[\Lambda] \geq \rho$. Hence, given event Λ, the entropy of the posterior that G has about Y is always lower bounded by (3.3.15). We thus obtain from (3.3.13) that

$$\mathbb{H}(Y \mid G) \geq \mathbb{P}[\Lambda]\, \mathbb{H}(Y \mid G, \Lambda) \geq \rho \mathbb{H}(Y) - (1-\rho) \log \frac{n}{1-\rho} + \rho \log \rho.$$

Finally, we obtain

$$\begin{aligned}
\mathbb{I}(G; Y) &= \mathbb{H}(Y) - \mathbb{H}(Y \mid G) \\
&\leq \mathbb{H}(Y) - \rho \mathbb{H}(Y) + (1-\rho) \log \frac{n}{1-\rho} + \rho \log \frac{1}{\rho} \\
&= (1-\rho)(\mathbb{H}(Y) + \log n) + (1-\rho) \log \frac{1}{1-\rho} + \rho \log \frac{1}{\rho}.
\end{aligned} \tag{3.3.16}$$

From now on, we assume that $k \geq 2$ (this is because the result of Lemma 3.6 is clear when $k = 1$). By the definition of I_0 in (3.3.16), we have $\frac{I_0}{S_{\min}} = \frac{\alpha}{7k \max\{\log n, \log \frac{1}{\alpha}\}} \leq \frac{1}{14}$ due to the fact that $\alpha \leq \log n$. By the definition of β in Eq. (3.3.7), we know that when $\frac{I_0}{S_{\min}} \leq \frac{1}{14}$, it holds that

$$1 - \rho = k(1 - \beta) = k\left(\frac{1}{2} - \frac{1}{2}\sqrt{1 - 4\frac{I_0}{S_{\min}}}\right) < \frac{11kI_0}{10S_{\min}}.$$

To prove this, one can show that the function $f(x) = \frac{\left(1/2 - (1/2)\sqrt{1-4x}\right)}{x}$ is monotone increasing for $x \in (0, 1/14]$. By plugging in $x = 1/14$ we obtain $f(1/14) \leq 11/10$. Combining the above equation with Eq. (3.3.6) we get $1 - \rho < \frac{11}{10}\frac{\alpha}{7\max\{\log n, \log \frac{1}{\alpha}\}} < \frac{11\alpha}{70 \log n}$. Now, from (3.3.16) we have

$$\begin{aligned}
\mathbb{I}(G; Y) &< \frac{11\alpha}{70}\frac{\mathbb{H}(Y) + \log n}{\log n} + (1-\rho)\log\frac{1}{1-\rho} + (1-(1-\rho))\log\frac{1}{1-(1-\rho)} \\
&\overset{(a)}{<} \frac{11\alpha}{35} + (1-\rho)\left(\log\frac{1}{1-\rho} + \frac{1}{\ln 2}\right) \\
&< \frac{11\alpha}{35} + \frac{11\alpha}{70}\frac{\log\frac{70}{11} + \log\left(\max\{\log n, \log(1/\alpha)\}\right) + \log\frac{1}{\alpha} + \frac{1}{\ln 2}}{\max\{\log n, \log(1/\alpha)\}} \\
&\overset{(b)}{\leq} \frac{11\alpha}{35} + \frac{11\alpha}{70}\left(\frac{\log\frac{70}{11} + \log\log n + \frac{1}{\ln 2}}{\log n} + 1\right) \overset{(c)}{<} \alpha.
\end{aligned}$$

Here, (a) follows from the fact that $\mathbb{H}(Y)$ is less than $\log n$ (because $|\mathcal{Y}| = n$), and the inequality $-(1-x)\log(1-x) < x/(\ln 2)$ for $x \in (0,1)$. Also, (b) follows from simple calculus steps which we omit for the sake of space, and (c) simply follows when $n \geq 3$. For $n = 2$, the proof of Lemma 3.5 can be done in a simpler way as above, and we relegate it to §A.1.3.

3.4 An Upper Bound on the Utility

In this section, we establish an upper bound on the ratio between the greedy policy and the optimal policy, which involves $S_{\min}$.

Theorem 3.9. *There exists an example, where* $\mathbb{I}\left(\pi_{\mathrm{MIS}[k']}; Y\right) \leq \mathbb{I}\left(\pi_{\mathrm{OPT}[k]}; Y\right) \cdot O(S_{\min})$.

In the following, we will provide an example which meets the upper bound in Theorem 3.9. We first describe the high-level intuition behind the example in §3.4.1, and then present details in §3.4.2.

3.4.1 A Bad Example for the MIS Criterion

In our example, we first design $T+1$ sets of tests $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{T+1}$ (the value of T will be specified later). By design, tests in these sets have low information gain on their own: in the beginning, all these tests have 0 outcomes with a high probability. However, if one test $v^{(i)} \in \mathcal{V}_i$ from set i is observed to have a positive outcome, then one can always find a test $v^{(i+1)} \in \mathcal{V}_{i+1}$, which is very informative about the remaining hypotheses that are consistent with the outcome of $v^{(i)}$. A smart policy will (sequentially) pick tests among these sets, and one can show that with at most $2T$ tests (see §3.4.2), this policy will reduce $\mathbb{H}(Y)$ by at least $(\log n)/T$ bits.

To "confuse" the greedy policy, we design another set of tests $\mathcal{U}$, with infinitely many noisy tests, and the deterministic outcome of each test takes value among $\{0,1\}$ with equal probability. As in Example 3.1, we assume that observed outcome of each test $Z \in \mathcal{Z}$ is perturbed by a binary symmetric channel with flip-over probability ϵ. If $S_{\min} = (1-2\epsilon)^2$ is sufficiently large (in fact, in our example, $S_{\min} \geq \Omega(1/\log n)$), one can show that with high probability, the greedy policy always picks among this set of tests, and within $2T$ tests, the gain of the greedy policy is at most $O(S_{\min}T)$ Bits. Setting $T = 2\sqrt{\log n}$, the ratio between the gain of greedy and the smarter policy is at most $c_0 S_{\min}$, where c_0 is some constant (note that when $S_{\min}$ is small we have $c_0 S_{\min} \approx (1 - \exp(c_0 S_{\min}))$).

3.4.2 Detailed Construction of the Treasure Hunt Example

Consider the following *treasure hunt* example. Assume that the hidden variable Y takes value among set $\mathcal{Y} = \{y_1, \cdots, y_n\}$ with uniform distribution. Define $T \triangleq 2\sqrt{\log n}$, and let $k' = k = 2T$, i.e., the greedy policy π_{MIS} and the optimal policy π_{OPT} have the same budget. We now design a problem with $\frac{2(10+2\log\log n)}{\log n} \leq S_{\min} \leq \frac{1}{256\sqrt{\log n}(\log\log n)^2}$, such that π_{MIS} performs considerably worse w.r.t. π_{OPT} (indeed, the ratio is a factor of $S_{\min}$). Finally, note that we choose n sufficiently large so that the bounds provided are meaningful.

The treasure hunt example. We first define $T+1$ types of tests, namely Type 1, Type 2, ..., Type $T+1$. The first type of tests, $\mathcal{V}_1$, contains a total number of T tests, all of which have binary outcomes. We denote the corresponding set of random variables by

$\mathbf{X}_{\mathcal{V}_1} = \{X_1^{(1)}, X_2^{(1)}, \dots X_T^{(1)}\}$. We partition the set $\mathcal{Y}$ into T equal-size groups, denoted by $\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_T$, each containing n/T values (assume n is such that this partition is possible). Each test of Type 1 can be thought to be informative about only a small fraction of $\mathcal{Y}_1$. Specifically, imagine that we further partition the set $\mathcal{Y}_1$ into T groups of equal size, denoted by $\mathcal{Y}_{1,1}, \mathcal{Y}_{1,2}, \cdots, \mathcal{Y}_{1,T}$. For each $i \in \{1, \dots, T\}$, we define

$$X_i^{(1)} = \begin{cases} 1, & \text{if } y \in \mathcal{Y}_{1,i}; \\ 0, & \text{o.w.} \end{cases}$$

The mutual information between each test $X_i^{(1)}$ and Y is

$$\begin{aligned} \mathbb{I}\left(X_i^{(1)}; Y\right) &= \mathbb{H}\left(X_i^{(1)}\right) - \mathbb{H}\left(X_i^{(1)} \mid Y\right) \\ &= \frac{1}{T^2} \log T^2 + \left(1 - \frac{1}{T^2}\right) \log\left(\frac{T^2}{T^2 - 1}\right) \\ &\leq \frac{\log T^2}{T^2} + \frac{2}{T^2} \\ &= \frac{4 + \log\log n}{4 \log n}. \end{aligned} \tag{3.4.1}$$

The second set of tests, $\mathcal{V}_2$, also contains T tests. Their corresponding outcomes are denoted by $\mathbf{X}_{\mathcal{V}_2} = \{X_1^{(2)}, X_2^{(2)}, \dots, X_T^{(2)}\}$. Each test in $\mathcal{V}_2$ has three possible outcomes. When $y \in \mathcal{Y}_{1,i}$, $X_i^{(2)}$ takes values among $\{1, 2\}$ with equal probability $1/2$; when $y \notin \mathcal{Y}_{1,i}$, $X_i^{(2)} = 0$. For each $i \in \{1, \dots, T\}$, we denote $\mathcal{Y}_{1,i,1}$ and $\mathcal{Y}_{1,i,2}$ to be the set of values of y on which $X_i^{(2)} = 1$ and $X_i^{(2)} = 2$, then

$$X_i^{(2)} = \begin{cases} 2, & \text{if } y \in \mathcal{Y}_{1,i,2}; \\ 1, & \text{if } y \in \mathcal{Y}_{1,i,1}; \\ 0, & \text{o.w.} \end{cases}$$

The mutual information between each test $X_i^{(2)}$ and Y is

$$\begin{aligned} \mathbb{I}\left(X_i^{(2)}; Y\right) &= \mathbb{H}\left(X_i^{(2)}\right) - \mathbb{H}\left(X_i^{(2)} \mid Y\right) \\ &= \frac{1}{2T^2} \log 2T^2 + \frac{1}{2T^2} \log 2T^2 + \left(1 - \frac{1}{T^2}\right) \log\left(\frac{T^2}{T^2 - 1}\right) \\ &\leq \frac{\log 2T^2}{T^2} + \frac{2}{T^2} \\ &= \frac{5 + \log\log n}{4 \log n}. \end{aligned} \tag{3.4.2}$$

Further, there are a total number of $2T$ tests of Type 3, with $\mathbf{X}_{\mathcal{V}_3} = \{X_1^{(3)}, X_2^{(3)}, \dots X_{2T}^{(3)}\}$. Each of the tests has 5 possible outcomes. Intuitively, we design these tests to further refine the set of values Y can take based on the outcome of tests in $\mathcal{V}_2$: if one of the tests in $\mathcal{V}_2$ has non-zero realization, then there exists a test $X_i^{(3)} \in \mathbf{X}_{\mathcal{V}_3}$ that will help us identify a much smaller subset of $\mathcal{Y}$. Formally, for $i \in \{1, \dots, T\}, j \in \{1, 2\}$, and $l \in \{1, 2, 3, 4\}$, we denote $\mathcal{Y}_{1,i,j,l}$ to be the set of values of y on which $X_{2i+1-j}^{(3)} = l$, and each $\mathcal{Y}_{1,i,j,l}$ contains $\frac{n}{4\times 2T^2}$ values. We define

$$X_{2i+1-j}^{(3)} = \begin{cases} 4, & \text{if } y \in \mathcal{Y}_{1,i,j,4}; \\ 3, & \text{if } y \in \mathcal{Y}_{1,i,j,3}; \\ 2, & \text{if } y \in \mathcal{Y}_{1,i,j,2}; \\ 1, & \text{if } y \in \mathcal{Y}_{1,i,j,1}; \\ 0, & \text{o.w.} \end{cases}$$

For $i \in \{1, \dots, 2T\}$, the mutual information between each test $X_i^{(3)}$ and Y is

$$\begin{aligned} \mathbb{I}\left(X_i^{(3)}; Y\right) &= 4 \times \frac{1}{4 \times 2T^2} \log\left(4 \times 2T^2\right) + \left(1 - \frac{1}{2T^2}\right) \log\left(\frac{2T^2}{2T^2 - 1}\right) \\ &\leq \frac{\log(4 \times 2T^2)}{2T^2} + \frac{2}{2T^2} \\ &= \frac{7 + \log\log n}{8 \log n}. \end{aligned} \tag{3.4.3}$$

Similarly, we define tests of Type t, $t \in \{2, \dots, T+1\}$ to be $\mathcal{V}_t$, with $|\mathcal{V}_t| = \prod_{i=1}^{t-2} 2^i$. Those tests, if sequentially performed, behave as follows: If one of the tests in $\mathcal{V}_{t-1}$ has non-zero realization, then one can perform a test in $X_i^{(t)} \in \mathbf{X}_{\mathcal{V}_t}$, and the outcome of this test can reduce the number of consistent hypotheses to a factor of $\frac{1}{2^{t-1}}$.

Suppose there is a "smart" policy, denoted by π^s, which works as follows. It first performs all the T tests in $\mathcal{V}_1$, and the probability that one of them has non-zero outcome is $1/T$. If this happens, then π^s sequentially picks T more tests from each of the sets $\mathcal{V}_2, \mathcal{V}_3, \dots, \mathcal{V}_{T+1}$. Test $e \in \mathcal{V}_i$ will reduce the number of valid values of $\mathcal{Y}$ by a factor of $\frac{1}{2^{i-1}}$. Hence, by noting $\frac{n}{T^2}\left(\frac{1}{2} \times \frac{1}{4} \times \cdots \times \frac{1}{2^{T+1}}\right) < 1$ we can see that, if $y \in \mathcal{Y}_1$, then after $2T$ tests, we get the right hypothesis (i.e., the policy π^s reduces $\mathbb{H}(Y \mid \pi)$ to 0). Since $y \in \mathcal{Y}_1$ occurs with probability $1/T$, we can bound the gain of π^s by

$$\mathbb{I}(\pi^s; Y) \geq (\log n)/T. \tag{3.4.4}$$

The greedy policy. We now define another set of tests and show that with high probability, the greedy policy prefers this set, but performing tests in this set gives a relatively low gain in terms of entropy reduction. Denote this set of tests by $\mathcal{V}_0$. There are infinitely many identical tests in $\mathcal{V}_0$. We denote the set of random variables by $\mathbf{X}_{\mathcal{V}_0} = \{X_1^{(0)}, X_2^{(0)}, X_3^{(3)}, \dots\}$.

For any fix Y, the observed outcome of any $X^{(0)} \in \mathbf{X}_{\mathcal{V}_0}$ is flipped from the (deterministic) outcome $D^{(0)}$ with probability ϵ (so that $S_{\min} = (1-2\epsilon)^2$), and the flipping events of the tests are independent (i.e., each test is associated with a binary symmetric noise channel). Assume that initially, the deterministic outcome $D_i^{(0)}$ of $X_i^{(0)}$ is uniformly distributed among $\{0,1\}$. In particular, let $\mathcal{Y}^1 \triangleq \mathcal{Y}_1 \cup \cdots \cup \mathcal{Y}_{T/2}$, and $\mathcal{Y}^0 \triangleq \mathcal{Y}_{T/2+1} \cup \cdots \cup \mathcal{Y}_T$, we define for each i,

$$D_i^{(0)} = \begin{cases} 1, & \text{if } y \in \mathcal{Y}^1; \\ 0, & \text{if } y \in \mathcal{Y}^0; \end{cases}$$

and the observed outcome $X_i^{(0)} = D_i^{(0)} \oplus N_i$, with $Pr(N_i = 1) = \epsilon$.

Then, it is easy to check that in the very beginning (where Y has a uniform distribution) we have $\mathbb{I}\left(X_i^{(0)}; Y\right) = 1 - h_2(\epsilon)$. We prove that the greedy policy π_{MIS} picks these tests with high probability. The following lemma characterizes such behavior of π_{MIS}.

Lemma 3.10. *Assume that* $\frac{2(10+2\log\log n)}{\log n} \le S_{\min} \le \frac{1}{256\sqrt{\log n}(\log\log n)^2}$. *With probability at least* $1 - 4\sqrt{\log n}\exp\left(-2(\log\log n)^2\right)$, $\pi_{\text{MIS}[2T]}$ *will pick* $2T$ *tests in* $\mathcal{V}_0$.

The proof of this lemma appears in §A.1.4. Now, note that $\mathbb{H}(Y)$ can at most be $\log n$ under any distribution. So the gain of π_{MIS} can be bounded from above as follows.

$$\mathbb{I}(\pi_{\text{MIS}}; Y) \le \mathbb{I}\left(X_1^{(0)}, \dots, X_{2T}^{(0)}; Y\right) + \left(4\sqrt{\log n}\exp\left(-2(\log\log n)^2\right)\right)\log n.$$

Let us now bound the mutual information term. We have

$$\begin{aligned}
\mathbb{I}\left(X_1^{(0)}, X_2^{(0)} \dots, X_{2T}^{(0)}; Y\right) &= \mathbb{H}\left(X_1^{(0)}, X_2^{(0)} \dots, X_{2T}^{(0)}\right) - \mathbb{H}\left(X_1^{(0)}, X_2^{(0)} \dots, X_{2T}^{(0)} \mid Y\right) \\
&\le \sum_{i=1}^{2T}\left(\mathbb{H}\left(X_i^{(0)}\right) - \mathbb{H}\left(X_i^{(0)} \mid Y\right)\right) \\
&= \sum_{i=1}^{2T}\mathbb{I}\left(X_i^{(0)}; D_i\right) \\
&= 2T(1 - h_2(\epsilon))
\end{aligned}$$

As a result, we can write

$$\begin{aligned}\mathbb{I}\left(\pi_{\mathrm{MIS}}; Y\right) &\leq 2T(1-h_2(\epsilon)) + 4\sqrt{\log n}\exp\left(-2(\log\log n)^2\right)\log n \\ &\leq 4TS_{\min} + 4\sqrt{\log n}\exp\left(-2(\log\log n)^2\right)\log n\end{aligned}$$

Combining the above Inequality with Eq. (3.4.4) we obtain

$$\begin{aligned}\frac{\mathbb{I}\left(\pi_{\mathrm{MIS}}; Y\right)}{\mathbb{I}\left(\pi^s; Y\right)} &\leq \frac{4TS_{\min} + 4\sqrt{\log n}\exp\left(-2(\log\log n)^2\right)\log n}{\frac{\log n}{T}} \\ &= 16S_{\min} + 8\log n v^{\left(-2(\log\log n)^2\right)} \\ &< 32S_{\min}.\end{aligned}$$

Hence, the gain of the greedy policy (when allowed to choose $2T$ tests) can be at most a fraction $32S_{\min}$ of the optimal policy which is also allowed to choose $2T$ tests.

Remark 3.11. Note that for our example to hold, we actually require that $S_{\min}$ to be at least $\Omega\left(1/\log n\right)$. Also, note that both $S_{\min}$ and $1/\log n$ are involved in the lower bound in Theorem 3.3. It remains an open problem to decide which combination of $S_{\min}$ and $1/\log n$ is indeed necessary for the lower bound.

3.5 Summary

In this chapter, we presented a theoretical analysis of the most informative selection policy for the sequential information maximization problem. We proved lower and upper bounds which relate the performance of the MIS policy to the performance of the optimal policy. In particular, our bounds show that the performance of the MIS policy is closely related to the noise level in the system: Under common assumptions made about the noise, the sequential information maximization criterion behaves near-optimally. Our results theoretically justify why the greedily maximizing mutual information has been found to be effective in theses settings. We further constructed an example where the greedy policy performed considerably worse compared with the optimal policy. Our analysis suggests that in such cases, the greedy approach could be misled to pick non-informative tests, and thus nonmyopic policies, e.g., using look-ahead, might be required.

4

Adaptive Information Acquisition for Decision Making

In §3, we have studied a basic version of the adaptive information acquisition problem, where the tests and the target random variable Y to be learned are modeled under the *Naïve Bayes* structure. This setting is useful when the goal of information gathering is to reduce the overall uncertainty in the system. However, in many practical applications, the collection of information is not a goal of its own, but rather a means for making informed decisions. In such cases, the tests which we can perform might *not* be *directly informative* about the target Y (which now denotes which decision to make). Hence, the conditional independence assumption no longer holds, and one must consider more complex probabilistic models and algorithmic frameworks for solving these problems.

Example 4.1. *As a running example, let us look at a more general variant of the medical diagnosis problem: A doctor can adaptively perform medical tests on a patient, each of which reveals some information about the patient's physical condition. Now, rather than learning the patient's exact physical condition, imagine that the goal for diagnosis is to predict the* ***best treatment****. We can assume that the outcomes of medical tests are conditionally independent given the patient's condition, but in general, they are not conditionally independent given the treatment, which is made based on the patient's condition. The challenging problem is to devise an adaptive policy by which the doctor can quickly make effective decisions.*

In this chapter, we focus on such challenging, yet practically relevant *decision-making* tasks, where we aim to a learn the value of some unknown target variable through a sequence of *correlated* and *noisy* tests. Here, by "correlated" we mean that test outcomes can be conditionally dependent given the hidden target random variable Y.

As introduced in §2.2.2, a natural formalism for this problem is by the *decision-theoretic* notion of *value of information* (c.f., Eq. (2.2.4)). It is known that common greedy heuristics, such as the myopic VoI policy (which greedily optimizes Eq. (2.2.4)), the MIS policy (which myopically maximizes the information gain w.r.t. the distribution over Y), and the generalized binary search policy (which greedily maximizes the reduction in the probability mass of the hypotheses) and can perform arbitrarily poorly [GKR10a]. Crucially, most of these adaptive policies have been designed for gathering information, but not for making decisions based on this information. In this chapter, we seek to tackle the decision-making problems, by investigating novel, efficient objectives which are amenable to greedy optimization.

DiRECt. We start by describing a novel formulation of the optimal value of information problem, and introduce DiRECt (see §4.3), an efficient, yet near-optimal algorithm for solving the nonmyopically optimizing value of information. Crucially, DiRECt uses a novel *surrogate objective* that is: (1) aligned with the value of information problem (2) efficient to evaluate and (3) *adaptive submodular*. This latter property enables us to utilize an efficient greedy optimization while providing strong approximation guarantees.

Efficient optimization In addition to pursuing algorithms with strong theoretical guarantees, we also seek to address the important practical issues that arise when deploying our algorithms. In fact, the class of submodular surrogate-based algorithms we consider may scale poorly when facing a large number of tests, because a vanilla implementation of these algorithms requires enumerating all possible realizations of the test outcomes, which, in the worst-case could be exponential in the number of tests. To alleviate this problem, we propose a novel sampling-based strategy (see §4.4), which allows efficient information gathering with strong theoretical guarantees. We show that with sufficient amount of samples, one can identify a near-optimal decision with high probability.

ECED. One catch for the DIRECT framework is that our theoretical guarantee only bounds the cost required to *perfectly determine* the optimal decision (i.e., with 100% accuracy). However, in practice, due to noise and budget constraint, we may want to stop performing more tests when we are confident enough. To accommodate such constraints posed by noise, we further introduce ECED (see §4.5), a novel algorithm for adaptively acquiring *decision-relevant* information. We prove that when the test outcomes are binary, and the noise on test outcomes are mutually independent, ECED is guaranteed to obtain near-optimal cost. We develop a theoretical framework for analyzing such sequential policies, where we leverage an information-theoretic auxiliary function to reason about the effect of noise and combine it with the theory of adaptive submodularity to attain the near-optimal bound. We demonstrate strong empirical performance for our proposed algorithms on several problem instances, including Bayesian experimental design for behavioral economics, interactive troubleshooting, active preference learning, and active touch-based localization.

Organization of this chapter. We begin by introducing basic notations and the formally stating the value of information problem in §4.1.1. In §4.1.2 we propose an equivalent formulation of the value of information problem, which we call the *Decision Region Determination* (DRD) problem. Most discussions in this chapter will be centered around it. In §4.2, we review existing methods for solving this problem. In §4.3, we present the DIRECT algorithm and discuss its theoretical guarantees. In §4.4, we investigate the practical aspects of our framework and propose an efficient sampling strategy when exact optimization of our surrogates is not feasible. We move on to the noisy setting in §4.5, and show how one can use ECED to handle noise in a principled manner. After presenting our theoretical contributions, we evaluate our algorithms in §4.6, and summarize this chapter in §4.7.

4.1 Problem Statement

We now state the general optimal value of information problem. Note in this chapter, we mainly look into this problem from the *adaptive minimum cost coverage* perspective (Problem 2.1.3), although some parts of our analysis framework build upon theoretical tools devised for adaptive submodular *maximization*.

We start with defining the optimal VoI problem in §4.1.1; then we state its DRD equivalence in §4.1.2.

4.1.1 The Optimal Value of Information Problem

Assume that there is some unknown hidden discrete random variable $\Theta \in \text{supp}(\Theta)$ upon which we want to make a decision. Here, $\text{supp}(\Theta)$ denote the support of the distribution on Θ. In our medical diagnostics example, Θ may represent the condition of the patient. We are given a set $\mathcal{V} = \{1, \ldots, t\}$ of possible tests; performing each test $v \in \mathcal{V}$ incurs a certain cost of $c(v) > 0$ and produces an outcome $X_v \in \mathcal{X}$ that is correlated with Θ. Further assume that there is a known prior distribution $\mathbb{P}[\Theta, X_1, \ldots, X_t]$ over the hidden variable and test outcomes admitting efficient inference. Crucially, we assume that X_v's are *conditionally independent* given the hidden state Θ, i.e.,

$$\mathbb{P}[\Theta, X_1, \ldots, X_t] = \mathbb{P}[\Theta] \prod_{v \in \mathcal{V}} \mathbb{P}[X_v \mid \Theta].$$

In this chapter, we assume that the parameters of the above distributions are given.

Suppose there is a finite set of decisions $\mathcal{Y} = \{y_1, \ldots, y_n\}$ that we can choose from. After performing a set of tests and observing their outcomes, we want to make the best decision given our belief about the hidden variable Θ (e.g., we must decide how to treat the patient). Hence, the value of Y depends on Θ. The probabilistic model is given in Fig. 4.1.

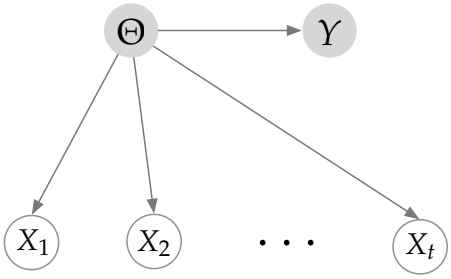


Figure 4.1: The graphical model.

Formally, as described in §2.2.2, we quantify the benefit of making a decision $y \in \mathcal{Y}$ for any $\theta \in \text{supp}(\Theta)$ by a utility function $u : \text{supp}(\Theta) \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$. We use $U(\cdot)$ to denote the expected value of a decision. The expected value of a decision y after observing the

outcomes $\mathbf{x}_\mathcal{A}$ of a set of tests $\mathcal{A}$ is

$$U(y \mid \mathbf{x}_\mathcal{A}) = \mathbb{E}_\theta[u(\theta, y) \mid \mathbf{x}_\mathcal{A}]\,.$$

The value of a particular set of observations $\mathbf{x}_\mathcal{A}$ is then defined as:

$$\mathrm{VoI}(\mathbf{x}_\mathcal{A}) = \max_{y \in \mathcal{Y}} U(y \mid \mathbf{x}_\mathcal{A}), \tag{4.1.1}$$

i.e., the maximum expected utility achievable when acting upon observations $\mathbf{x}_\mathcal{A}$.

Consider performing *all* tests, receiving outcomes $\mathbf{x}_\mathcal{V}$, and making the most informed decision possible. This would achieve a value of $\mathrm{VoI}(\mathbf{x}_\mathcal{V})$. However, it may be possible to reach nearly $\mathrm{VoI}(\mathbf{x}_\mathcal{V})$ with far fewer tests (see Fig. 4.2a for illustration). Our goal is to adaptively select the cheapest tests to do so. Formally, we define the *regret* of a decision y given observations $\mathbf{x}_\mathcal{A}$ by

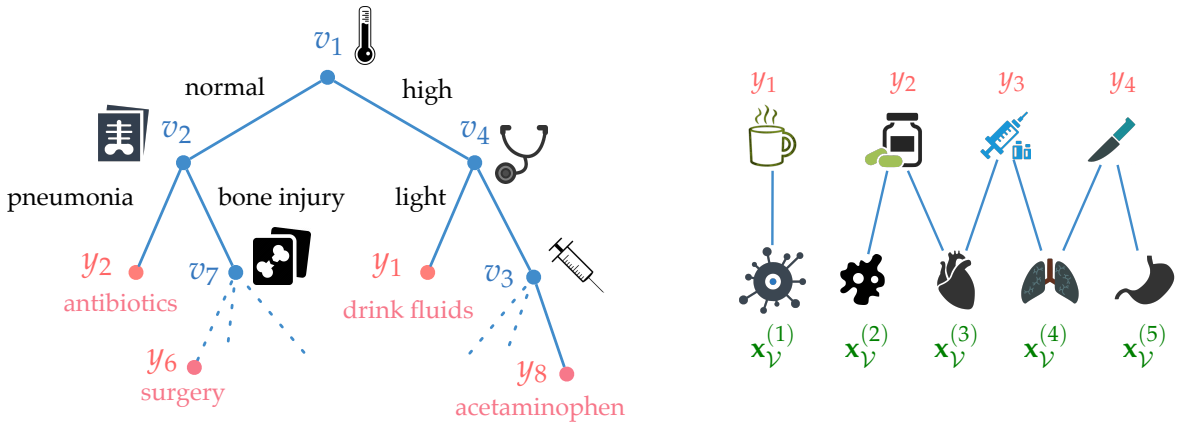
$$R(y \mid \mathbf{x}_\mathcal{A}) = \max_{\mathbf{x}_\mathcal{V} : \mathbb{P}[\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{A}] > 0} [\mathrm{VoI}(\mathbf{x}_\mathcal{V}) - U(y \mid \mathbf{x}_\mathcal{V})].$$

Clearly, this regret is also an upper bound on the *expected* loss in expected utility (if we stop upon observing $\mathbf{x}_\mathcal{A}$ and committing to action y), i.e., $\mathbb{E}_{\mathbf{x}_\mathcal{V}}[\mathrm{VoI}(\mathbf{x}_\mathcal{V}) - U(y \mid \mathbf{x}_\mathcal{V}) \mid \mathbf{x}_\mathcal{A}]$. Our goal is to find a policy of minimum cost with low regret. Formally, for some fixed small tolerance $\varepsilon \geq 0$, we seek a policy π^* with minimum cost, such that upon termination, π^* will suffer regret of at most ε:

$$\begin{aligned} &\pi^* \in \arg\min_\pi \mathrm{cost}(\pi), \text{ s.t.} \\ &\forall \mathbf{x}_\mathcal{V}\ \exists y : R(y \mid \mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) \leq \varepsilon \text{ whenever } \mathbb{P}\left[\mathbf{x}_\mathcal{V}\right] > 0. \end{aligned} \tag{4.1.2}$$

where recall (from §2.1.2) that $\mathcal{S}(\pi, \mathbf{x}_\mathcal{V}) \subseteq \mathcal{V} \times \mathcal{X}$ denotes the set of observations obtained by running policy π until termination (likely before exhausting all tests) when the tests are in state $\mathbf{x}_\mathcal{V}$, and $\mathrm{cost}(\pi)$ may either denote the average cost or the worst-case cost of a policy. We will instantiate the $\mathrm{cost}(\cdot)$ function when we present our algorithms.

In other words, we require that each feasible policy satisfies the following condition: Upon termination, we must be able to commit to a decision, such that we lose *at most* ε expected utility, compared to the optimal decision we could have made if we had also observed *all remaining* unobserved variables. (see Fig. 4.2b for an illustration of the ε-optimal decisions for different observations.) We call Problem 4.1.2 the *nonmyopic value of information problem for achieving near-maximal utility (NVOI-NMU)*.



(a) Adaptive policy for optimal treatment

(b) Treatments and patient's conditions

Figure 4.2: Illustration of an adaptive policy for learning the optimal treatment in medical diagnosis. The bipartite graph on the right represents the mapping from patient's conditions (i.e., root-causes) to their near-optimal treatments (i.e., the utility of performing the treatment is at most ε away from the maximal utility for the corresponding root-cause).

Remark 4.2. In classical value of information, costs and utilities have the same units, and the goal is to maximize benefit minus cost. In many cases (e.g. medical diagnosis), this is not the case, so we formulate our problem to achieve near-maximal utility with minimum cost.

4.1.2 Decision Region Determination

We now state an equivalent formulation of the NVOI-NMU problem, which we call the *Decision Region Determination* (DRD) problem. In DRD, we are given

1. a set of *hypotheses* $\mathcal{H} := \{h_1, \ldots, h_s\}$;
2. a random variable H distributed over $\mathcal{H}$ with known distribution $\mathbb{P}$;
3. a set of tests $\mathcal{V} = \{1, \ldots, t\}$ modeled as *deterministic* functions $f_1, \ldots f_t : \mathcal{H} \to \mathcal{X}$;
4. a cost function $c : \mathcal{V} \to \mathbb{R}_{\geq 0}$ and
5. a collection of subsets $\mathcal{R}_{y_1}, \ldots, \mathcal{R}_{y_n} \subseteq \mathcal{H}$ called *decision regions*.

We seek a policy π^* of minimum cost (which adaptively picks tests v, observes their outcomes $X_v = f_v(H)$, where $H \in \mathcal{H}$ is the unknown hypothesis), such that upon

termination, there exists at least one decision region that contains all hypotheses consistent with the observations made by the policy. That is, we seek

$$\pi^* \in \arg\min_{\pi} \text{cost}(\pi), \text{s.t. } \forall h \; \exists y : \mathcal{H}(\mathcal{S}(\pi, h)) \subseteq \mathcal{R}_y. \tag{4.1.3}$$

Hereby $h \in \mathcal{H}$, and

$$\mathcal{H}(\mathbf{x}_\mathcal{A}) \triangleq \{h' \in \mathcal{H} : (i, x) \in \mathbf{x}_\mathcal{A} \Rightarrow f_i(h') = x\} \tag{4.1.4}$$

is the set of hypotheses consistent with $\mathbf{x}_\mathcal{A}$. To reduce the NVOI-NMU Problem (4.1.2) to DRD (4.1.3), we interpret every outcome vector $\mathbf{x}_\mathcal{V} \in \mathcal{X}^\mathcal{V}$ with positive probability as a hypothesis h. The interpretation of the prior, tests and costs follow immediately. Note that in DRD we do not have an explicit notion of "root-cause" as is in NVOI-NMU. It remains to define the decision regions. For each decision y, we set $\mathcal{R}_y$ to be the set of outcome vectors, for which y is an ε-optimal action, or formally:

$$\mathcal{R}_y \triangleq \{\mathbf{x}_\mathcal{V} : U(y \mid \mathbf{x}_\mathcal{V}) \geq \text{VoI}(\mathbf{x}_\mathcal{V}) - \varepsilon\}.$$

Fig. 4.3 illustrates the relation between decision regions $\mathcal{R}_y$'s (which are indexed by y's) and hypotheses h's (or equivalently, the outcomes $\mathbf{x}_\mathcal{V}$'s) with the medical diagnosis example (c.f. Example 4.1).

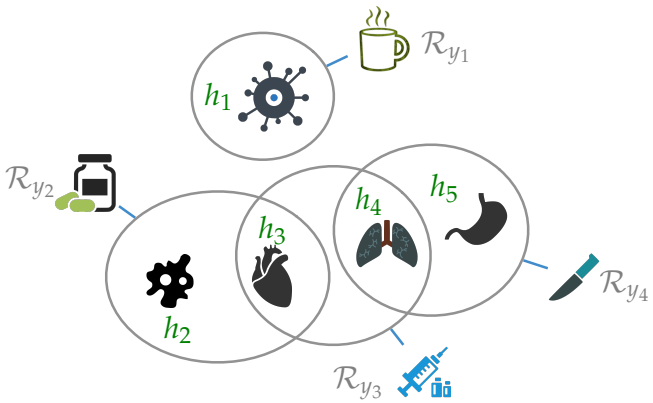


Figure 4.3: Hypotheses and decision regions drawn from the bipartite graph in Fig. 4.2b.

We summarize the (bijective) mapping from NVOI-NMU to DRD in Table 4.1.

Remark 4.3. In Problem 4.1.3, we require that test outcomes $\mathbf{X}_\mathcal{V}$ are deterministic given any hypothesis $h \in \mathcal{H}$. The prior distribution on H is, in fact, $\mathbb{P}[X_1, \ldots, X_t] = \sum_{\theta \in \text{supp}(\Theta)} \mathbb{P}[\Theta = \theta, X_1, \ldots, X_t]$. Thus, to construct the DRD problem from NVOI-NMU, we can simply enumerate all test outcomes (with non-zero probabilities) as our hypotheses, after which we only need to deal with deterministic distributions.

	NVOI-NMU	DRD
tests	test $v \in \mathcal{V}$	test $v \in \mathcal{V}$
test outcomes	full realization: $\mathbf{x}_{\mathcal{V}} \in \mathcal{X}^{\mathcal{V}}$	hypothesis: $h \in \mathcal{H}$
target decision	decision: $y \in \mathcal{Y}$	decision region: $\mathcal{R}_y \subseteq \mathcal{H}$
latent variable	root-cause: θ	-

Table 4.1: Mapping from NVOI-NMU to DRD.

Remark 4.4. In the general (non-realizable/noisy) case, i.e., $\mathbb{P}[X_v = x_v \mid \Theta] \in [0,1]$, each root-cause corresponds to a set of hypotheses with non-zero probabilities $\{h : \mathbb{P}[H = h \mid \theta] > 0\}$. In the realizable case, i.e., when $\mathbb{P}[X_v = x_v \mid \Theta] \in \{0,1\}\ \forall v \in \mathcal{V}, x_v \in \mathcal{X}$, it holds that $|\{h : \mathbb{P}[H = h \mid \theta] > 0\}| = 1\ \forall \theta \in \text{supp}(\Theta)$, meaning that each root-cause in NVOI-NMU corresponds to (exactly) one hypothesis in DRD.

4.2 Existing Approaches

In this section, we review some existing approaches for solving the NVOI-NMU/DRD problem, and provide problem instances showing why these approaches do not always work well.

4.2.1 Posterior-based Approaches

There is a large family of adaptive greedy policies (including the popular MIS policy π_{MIS} (as considered in §3), and the myopic VoI policy π_{VoI}), which use only information about the posterior target variable distribution to select the next test. These approaches are known as the *posterior-based approaches* [GKR10a]. More precisely, these policies define an objective/score function $\Delta(\cdot)$ which maps distributions over Y to real numbers, and at each time step select the test v which maximizes the score of the posterior distribution over y generated by adding x_v to the previously seen test outcomes. The most informative policy is posterior–based, as it scores each test using -1 times the entropy of the posterior distribution of Y:

$$\Delta_{\text{MIS}}(v \mid \mathbf{x}_{\mathcal{A}}) := \mathbb{H}(Y \mid \mathbf{x}_{\mathcal{A}}) - \mathbb{E}_{x_v}[\mathbb{H}(Y \mid \mathbf{x}_{\mathcal{A}}, x_v) \mid \mathbf{x}_{\mathcal{A}}]\,. \tag{4.2.1}$$

Likewise, the myopic value of information policy is also posterior–based, as it scores each test based on its expected gain in the value function, which is based on the posterior distribution on Y:

$$\Delta_{\text{VoI}}(v \mid \mathbf{x}_{\mathcal{A}}) := \mathbb{E}_{x_v}\Big[\text{VoI}(\mathbf{x}_{\mathcal{A}\cup\{v\}}) \mid \mathbf{x}_{\mathcal{A}}\Big] - \text{VoI}(\mathbf{x}_{\mathcal{A}}), \tag{4.2.2}$$

where $\text{VoI}(\cdot)$ is defined according to Eq. (4.1.1).

One can construct problem instances where the posterior-based approaches may fail (in the sense that they require cost which is exponentially higher than the optimal policy). In particular, Golovin, Krause, and Ray [GKR10a] show that

Theorem 4.5 (Theorem 9 of Golovin, Krause, and Ray [GKR10a])**.** *There exists a family of* DRD *instances with uniform priors such that the expect cost* $\text{cost}_{avg}(\pi) = \Omega\left(\frac{s}{\log s}\right)\text{cost}_{avg}(\pi^*)$ *for any posterior–based policy* π*, where* s *is the number of hypotheses and* π^* *is an optimal policy.*

In the following, we provide a *treasure-hunt* example that exhibits the above property.

A bad example for the posterior-based policies. Consider the problem instance illustrated in Fig. 4.4. Fix $q > 0$ to be some integer, and let $n = |\mathcal{Y}| = 2^q$. For each target value $y_i \in \mathcal{Y}$, there exist two hypotheses, i.e., $h_{i,1}$, $h_{i,0}$, such that $h_{i,1}, h_{i,0} \in \mathcal{R}_{y_i}$. Denote a hypothesis as $h_{i,x}$, if it belongs to the region of y_i and is indexed by x. We assume a uniform prior over the hypotheses: $\{h_{i,x}\}_{i\in\{1,\ldots,t\},x\in 0,1}$.

Figure 4.4: A DRD problem instance with non-overlapping decision regions.

There are three set of tests, and all of them have binary outcomes and unit cost (i.e., $\mathcal{X} = \{0, 1\}$, and $c(v) \equiv 1$). The first set $\mathcal{V}_1 := \{v_0\}$ contains one test v_0, which tells us the value of x of the underlying hypothesis $\theta_{i,x}$. Hence for all i, $H = h_{i,x} \Rightarrow X_{v_0} = x$. The second set of tests are designed to help us quickly discover the index of the target value via binary search if we have already run v_0, but to offer no information whatsoever (in terms of expected reduction in the prediction error, or expected reduction in entropy of Y) if v_0 has not yet been run. There are a total number of q tests in the second set

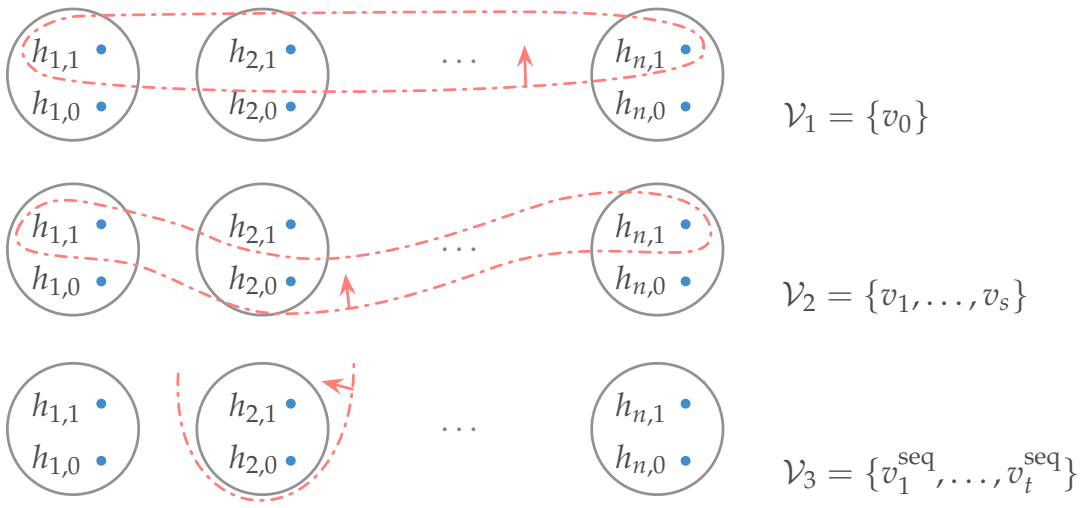


Figure 4.5: Illustration of the treasure hunt example where posterior-based policies fail.

$\mathcal{V}_2 := \{v_1, v_2, \ldots, v_q\}$. For $z \in \{1, \ldots, n\}$, let $b_k(z)$ be the k^{th} least-significant bit of the binary encoding of z, so that $z = \sum_{k=1}^{q} 2^{k-1} b_k(z)$. Then, if $H = h_{i,o}$, then the outcome of test $v_k \in \mathcal{V}_2$ is $X_k = \mathbb{1}\{\phi_k(i) = o\}$. The third set of tests are designed to allow us to do a (comparatively slow) sequential search on the index of the the target values. Specifically, we have $\mathcal{V}_3 := \{v_1^{\text{seq}}, \ldots, v_n^{\text{seq}}\}$, such that $H = h_{i,x} \Rightarrow X_{v_k^{\text{seq}}} = \mathbb{1}\{i = k\}$. We illustrate these three sets of tests in Fig. 4.5.

Now consider running a posterior-based policy, say, π_{MIS} (the same analysis also applies to the myopic value of information policy, which we omit from the paper). Note that in the beginning, no single test from $\mathcal{V}_1 \cup \mathcal{V}_2$ results in any change in the distribution over Y, as it remains uniform no matter with test is performed. Hence, the maximal informative policy only picks tests from $\mathcal{V}_3$, which have non-zero (positive) expected reduction in the posterior entropy of Y. In the likely event that the test chosen is not the index of Y, we are left with a residual problem in which tests in $\mathcal{V}_1 \cup \mathcal{V}_2$ still have no effect on the posterior. The only difference is that there is one less class, but the prior remains uniform. Hence our previous argument still applies, and π will repeatedly select tests in $\mathcal{V}_3$, until a test has an outcome of 1. In expectation, the cost of π is least $\text{cost}(\pi_{\text{MIS}}) \geq \frac{1}{n} \sum_{z=1}^{n} z = \frac{n+1}{2}$.

On the other hand, a smarter policy π^* will select test $v_0 \in \mathcal{V}_1$ first, and then performs a binary search by running test $v_1, \ldots, v_s \in \mathcal{V}_2$ to determine $b_k(i)$ for all $1 \leq k \leq s$ (and hence to determine the index i of Y). Since the tests have unit cost, the cost of π^* is $\text{cost}(\pi^*) = q + 1$.

Since $n = 2^q$, and $s = 2n = 2^{q+1}$, we conclude that

$$\text{cost}(\pi_{\text{MIS}}) = \frac{n+1}{2} > \frac{n}{2} = \frac{s}{4}\frac{q+1}{\log s} = \frac{s}{4\log(s)}\,\text{cost}(\pi^*).$$

4.2.2 Submodular Surrogates

An alternative approach to the posterior-based methods is to work directly with the probability mass of the hypotheses (instead of the *normalized* posterior probabilities). In the following, we review a class of objective functions that exhibit *adaptive submodularity*, which we call *submodular surrogates*, and elaborate on the pros and cons of the greedy policies based on those objectives.

Remark 4.6. It is a common strategy in machine learning to optimize surrogate objective functions. For example, consider a supervised learning task where we want to minimize classification error. Often we don't minimize misclassification error directly; instead, we optimize some surrogate, such as the hinge loss or log-loss. The approaches we discuss in this section is similar in spirit.

GBS: Generalized Binary Search

We first consider GBS, a greedy policy that generalizes the classical *binary search* algorithm. To be consistent with the active learning literature, we call $\mathcal{H}(\mathbf{x}_\mathcal{A})$ (as defined in Eq (4.1.4)) the *version space* that is consistent with observation $\mathbf{x}_\mathcal{A}$. In the Bayesian setting, GBS myopically attempts to shrink a measure of the version space (i.e., the cardinality or the probability mass) as quickly as possible. Formally, the GBS objective function is defined as

$$f_{\text{GBS}}(\mathbf{x}_\mathcal{A}) := \sum_{h\in\mathcal{H}} \mathbb{P}\left[h, \mathbf{x}_\mathcal{A}\right].$$

It thus scores a test by

$$\Delta_{\text{GBS}}(v \mid \mathbf{x}_\mathcal{A}) := \sum_{h\in\mathcal{H}} \mathbb{P}\left[h, \mathbf{x}_\mathcal{A}\right] - \sum_{x_v} \mathbb{P}\left[x_v \mid \mathbf{x}_\mathcal{A}\right] \sum_{h\in\mathcal{H}} \mathbb{P}\left[h, \mathbf{x}_\mathcal{A}, x_v\right],$$

and picks the one that maximizes the $\Delta_{\text{GBS}}(v \mid \mathbf{x}_\mathcal{A})/c(v)$. It can be shown that the function f_{GBS} is *adaptive submodular*. From Theorem 2.7 we know that in the realizable case, GBS achieves $O\left(\frac{1}{\min_h \mathbb{P}[h]}\right)$ approximation guarantee, if the goal is to identify the

true *hypothesis*. Also, if $\min_h \mathbb{P}[h]$ is sufficiently small, running GBS on a modified prior $p(h) \propto \max\left\{\mathbb{P}[h], \frac{1}{|\mathcal{H}|^2}\right\}$ improves the approximation factor to $O(\log |H|)$ [KPB99b; GK11b].

The above result is useful if each region $\mathcal{R}_y$ in the DRD problem (4.1.3) contains exactly *one* hypothesis. This special case is also known as the Optimal Decision Tree (ODT) problem [GG74; Ark+93; Gup+10; GK11a]. For general DRD problems, unfortunately, due to the lack of consideration of decision regions, GBS can perform quite badly. We provide a problem instance below to demonstrate when GBS may fail.

A bad example for GBS. Assume we are given a uniform prior over s hypotheses, $h_1, \ldots, h_s$, and two target decision regions $h_1, \ldots, h_{s-1} \in \mathcal{R}_{y_1}$, and $h_s \in \mathcal{R}_{y_2}$. There are s tests $\mathcal{V} = \{1, \ldots, s\}$ such that $\mathbb{P}[X_v = 1 \mid h_i] = \mathbb{1}\{i = v\}$ (all of unit cost). Here, $\mathbb{1}\{\cdot\}$ is the indicator function. See Fig. 4.6 for illustration.

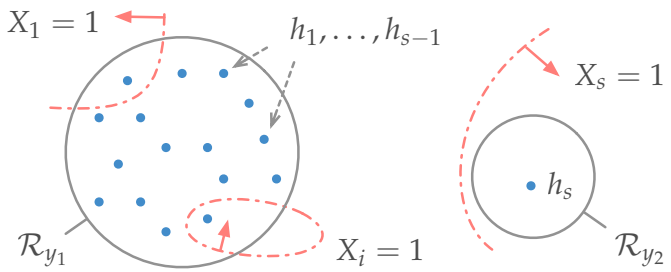


Figure 4.6: Example problem instance where GBS performs badly.

In this case, the optimal policy only needs to select test s. However GBS may choose tests $\{1, \ldots, s\}$ in order until running test v, where $H = h_v$ is the true root-cause. Given our uniform prior, it takes $s/2$ tests in expectation until this happens, so that GBS pays, in expectation, $s/2$ times the optimal expected cost in this instance.

EC^2: Equivalence Class Edge Cutting

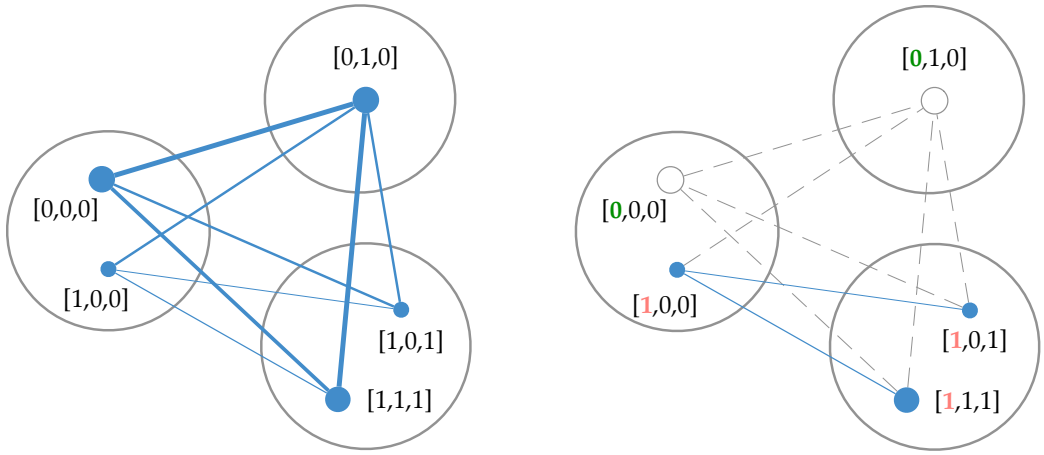
Now let us consider a special case of the DRD problem, where the decision regions are *disjoint*, i.e., $\mathcal{R}_{y_i} \cap \mathcal{R}_{y_j} = \emptyset$ for $i \neq j$. This means that each hypothesis h is associated with a unique decision. Such problem is known as the *Equivalence Class Determination* (ECD) problem [GKR10a].

Golovin, Krause, and Ray [GKR10a] proposes an algorithm, namely, *Equivalence Class Edge Cutting* (EC^2), which considers hypotheses as nodes in a graph $\mathcal{G}_{EC} = (\mathcal{H}, \mathcal{E})$, and

defines weighted edges between hypotheses in different decision regions (see Fig. 4.7a):

$$\mathcal{E} = \cup_{i \neq j} \{(h, h') : h \in \mathcal{R}_{y_i}, h' \in \mathcal{R}_{y_j}\}$$

The weight of an edge is defined as $w((h, h')) = \mathbb{P}[h] \cdot \mathbb{P}[h']$; similarly, the weight of a set of edges is $w(\mathcal{E}') = \sum_{(h,h') \in \mathcal{E}'} w((h, h'))$. An edge is consistent with the observation iff both hypotheses incident to the edge are consistent. Hence, a test v with outcome x_v is said to cut edges $\mathcal{E}(x_v) = \{(h, h') \in \mathcal{E} : f_v(h) \neq x_v \vee f_v(h') \neq x_v\}$ (see Fig. 4.7b). Performing tests will cut edges inconsistent with the observed test outcomes, and we aim to eliminate all inconsistent edges (until there is a single equivalence class/decision region left), while minimizing the expected cost incurred.



(a) Initializing the ECD graph $\mathcal{G}_{EC}$ to be cut

(b) Observing $X_1 = 1$.

Figure 4.7: Illustration of the equivalence class edge cutting algorithm. Hypotheses are represented in dots. The size of a dot is proportional to its probabilities.

The EC^2 objective is defined as the total weight of edges cut:

$$f_{\text{EC}^2}(\mathbf{x}_\mathcal{A}) := w\Big(\bigcup_{v \in \mathcal{A}} \mathcal{E}(x_v)\Big). \tag{4.2.3}$$

We observe that for solving the DRD problem, we can group together all hypotheses that share the same region assignments. Let $\mathbb{P}\left[\mathcal{R}_{y_i}\right]$ be the total prior probability mass of all hypotheses h in $\mathcal{R}_{y_i}$. Then the weight of edges between distinct decision regions $\mathcal{R}_{y_i}, \mathcal{R}_{y_j}$ is $w(\mathcal{R}_{y_i} \times \mathcal{R}_{y_j}) = \sum_{h \in \mathcal{R}_{y_i}, h' \in \mathcal{R}_{y_j}} \mathbb{P}[h]\,\mathbb{P}[h'] = \mathbb{P}\left[\mathcal{R}_{y_i}\right] \mathbb{P}\left[\mathcal{R}_{y_j}\right]$.
Naively, computing the total edge weight requires enumerating all pairs of regions. However, we can compute this in *linear* time by noting it is equivalent to an *elementary*

symmetric polynomial of degree 2: $\sum_{i \neq j} w(\mathcal{R}_{y_i} \times \mathcal{R}_{y_j}) = \frac{1}{2}\left(\left(\sum_i \mathbb{P}\left[\mathcal{R}_{y_i}\right]\right)^2 - \sum_i \mathbb{P}\left[\mathcal{R}_{y_i}\right]^2\right)$. We similarly compute the total edge weight after observations $\mathbf{x}_\mathcal{A}$ using $\mathbb{P}\left[\mathcal{R}_{y_i} \cap \mathcal{H}(\mathbf{x}_\mathcal{A})\right]$ for the probability mass of all hypotheses in $\mathcal{R}_{y_i}$ consistent with observations $\mathbf{x}_\mathcal{A}$. Finally, we subtract these two quantities to compute

$$f_{\text{EC}^2}(\mathbf{x}_\mathcal{A}) = \sum_{i \neq j} w(\mathcal{R}_{y_i} \times \mathcal{R}_{y_j}) - \sum_{i \neq j} w(\mathcal{R}_{y_i} \cap \mathcal{H}(\mathbf{x}_\mathcal{A}) \times \mathcal{R}_{y_j} \cap \mathcal{H}(\mathbf{x}_\mathcal{A})).$$

EC^2 scores a test by its expected marginal gain in f_{EC^2}, and at each round picks the one that maximizes the benefit-cost ratio. It can be shown that f_{EC^2} is adaptive submodular and strongly adaptive monotone (c.f. Definition 2.5) [GK11a], and hence EC^2 is near-optimal for the ECD problem.

HEC: Hyperedge Cutting

EC^2 crucially relies on the fact that decision regions are *disjoint*. In the presence of overlapping regions, there is no principled way to apply EC^2. Recently, we propose a natural generalization of EC^2, which we call the *Hyperedge Cutting* algorithm (HEC) [Jav+14], and prove that it can near-optimally solve the general DRD problem.

Different from EC^2 which deals with the ECD graph $\mathcal{G}_{EC}$ (e.g., Fig. 4.7a), the construction of HEC is based upon an alternate representation – a *hypergraph* $\mathcal{G}_{HEC} := (\mathcal{H}, \mathcal{E})$ for splitting decision regions. Observing certain test outcomes corresponds to down weighting or cutting *hyperedges* in this hypergraph. Here, our hyperedges are not sets, but *multisets*, a generalization of sets where members are allowed to appear more than once. As a result, a node can potentially appear in a hyperedge multiple times. See Fig. 4.8b for illustration.

In HEC, the set of hyperedges $\mathcal{E}$ consists of all multisets E of precisely k hypotheses, $E = \{h_1, \ldots, h_k\}$, such that a single decision region does not contain them all:

$$\mathcal{E} = \{E : |E| = k \wedge \nexists\, \mathcal{R} \text{ s.t. } \forall h \in E, h \in \mathcal{R}\}.$$

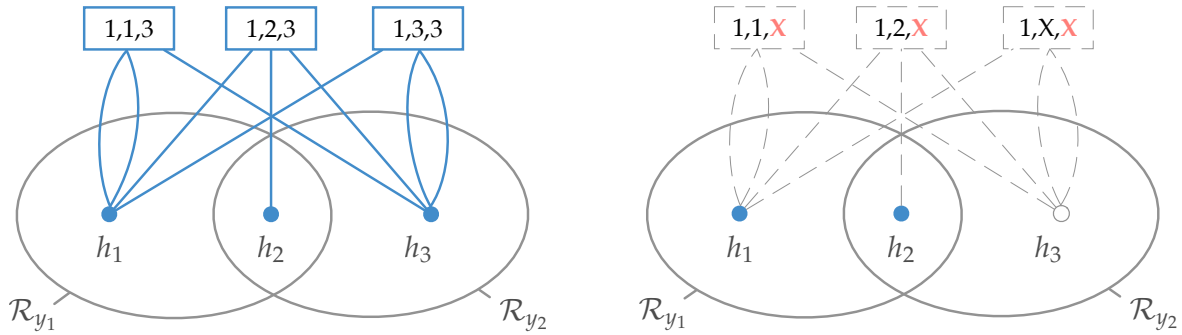
The key to attaining our results is the proper selection of hyperedge cardinality k. If k is too small, we cannot use solve the DRD problem; if k is too large, we waste computational effort, and our theoretical bounds loosen. Let us use ζ_y to denote the number of partitions, or *subregions* of decision region $\mathcal{R}_y$, where a subregion represents

a group of hypotheses which share the same region assignments. Then, a practical choice of the cardinality k is

$$k = \min\left(\max_{h\in\mathcal{H}} |\{\mathcal{R}: h \in \mathcal{R}\}|, \max_{y\in\mathcal{Y}} \zeta_y\right) + 1. \tag{4.2.4}$$

Hereby, $\max_h |\{\mathcal{R}: h \in \mathcal{R}\}|$ is the maximum number of regions that a hypothesis can be in, and $\max_{y\in\mathcal{Y}} \zeta_y$ is the maximum number of partitions in any region.

As with EC^2, we define the weight of a hyperedge E as $w(E) = \prod_{h\in E} \mathbb{P}[h]$; the weight of a set of edges is $w(\mathcal{E}') = \sum_{E\in\mathcal{E}'} w(E)$. A test v with outcome x_v is said to cut hyperedges $\mathcal{E}(x_v) = \{E \in \mathcal{E} : \exists h \in E, \text{ s.t. } f_v(h) \neq x_v\}$ (see Fig. 4.8b). We aim to eliminate all inconsistent edges (until there is a single equivalence class/decision region left), while minimizing the expected cost incurred.



(a) Initializing the hypergraph $\mathcal{G}_{HEC}$

(b) Observing h_3 to be inconsistent.

Figure 4.8: Illustration of the hyperedge cutting algorithm.

Following the definition of the EC^2 objective, the HEC objective is defined as the total weight of hyperedges cut:

$$f_{\text{HEC}}(\mathbf{x}_{\mathcal{A}}) := w\Big(\bigcup_{v\in\mathcal{A}} \mathcal{E}(x_v)\Big). \tag{4.2.5}$$

Theorem 4.7. *The objective function f_{HEC} defined in Eq. (4.2.5) is adaptive submodular and strongly adaptive monotone.*

By Theorem 4.7 and Theorem 2.7, we obtain the following result for our HEC Algorithm:

Theorem 4.8. *Assume a rational prior on the hypotheses. Let k be the hyperedge cardinality, π^* be the optimal policy, and $p_{\min} = \min_{h\in\mathcal{H}} P(h)$. Then, the performance of π_{HEC} is bounded by: $\text{cost}(\pi_{HEC}) \leq (k \ln(1/p_{\min}) + 1)\,\text{cost}(\pi^*)$.*

Note that as a consequence of adaptive submodularity, HEC also enjoys approximation guarantees against worst-case realizations of tests.

Remark 4.9. For the special case of disjoint regions (i.e., the ECD Problem, corresponding to $k = 2$), our objective f_{HEC} is equivalent to f_{EC^2}, and hence our Theorem 4.8 strictly generalizes the results of Golovin, Krause, and Ray [GKR10a]. Furthermore, in the special case where each region contains exactly one hypothesis ($k = 1$), and each test can have at most two outcomes, HEC is equivalent to the GBS algorithm and recovers its approximation guarantee.

Remark 4.10. The computational bottleneck for HEC lies in the construction of this hypergraph. In the worst case, this algorithm still has complexity $O\left(|\mathcal{H}|^k\right)$ (in principle, one can reduce it to $O\left((\text{number of subregions})^k\right)$ by viewing each subregion as a node in the hypergraph; however it is still exponential in k). This occurs when many, at least k, subregions share a single region. The complexity is then controlled by how many distinct subregions a single region can be shattered into, and the largest number of regions a single hypothesis can belong to. Thus, when we have large overlap between regions – the common case for NVOI-NMU, in particular with larger ε – HEC becomes infeasible.

4.3 The Decision Region Edge Cutting Algorithm

We now present *Decision Region Edge Cutting* (DIRECT), an *efficient* and *near-optimal* algorithm for solving the DRD problem.

4.3.1 The Noisy-OR Construction

Let $n = |\mathcal{Y}|$ be the number of decisions in $\mathcal{Y}$. Our strategy is to reduce the DRD problem to $O(n)$ instances of the ECD problem, such that solving *any one of them* is sufficient for solving the DRD problem. Crucially, the problem we end up solving depends on the unknown hypothesis h^*. We design our surrogate DIRECT so that it adaptively determines which instance to solve to minimize the expected total cost.

Concretely, we construct n different subproblem instances, one for each decision. The role of subproblem i is to determine whether the unknown hypothesis h^* is contained in decision region $\mathcal{R}_{y_i}$ or not. Thus we aim to distinguish all the hypotheses in this

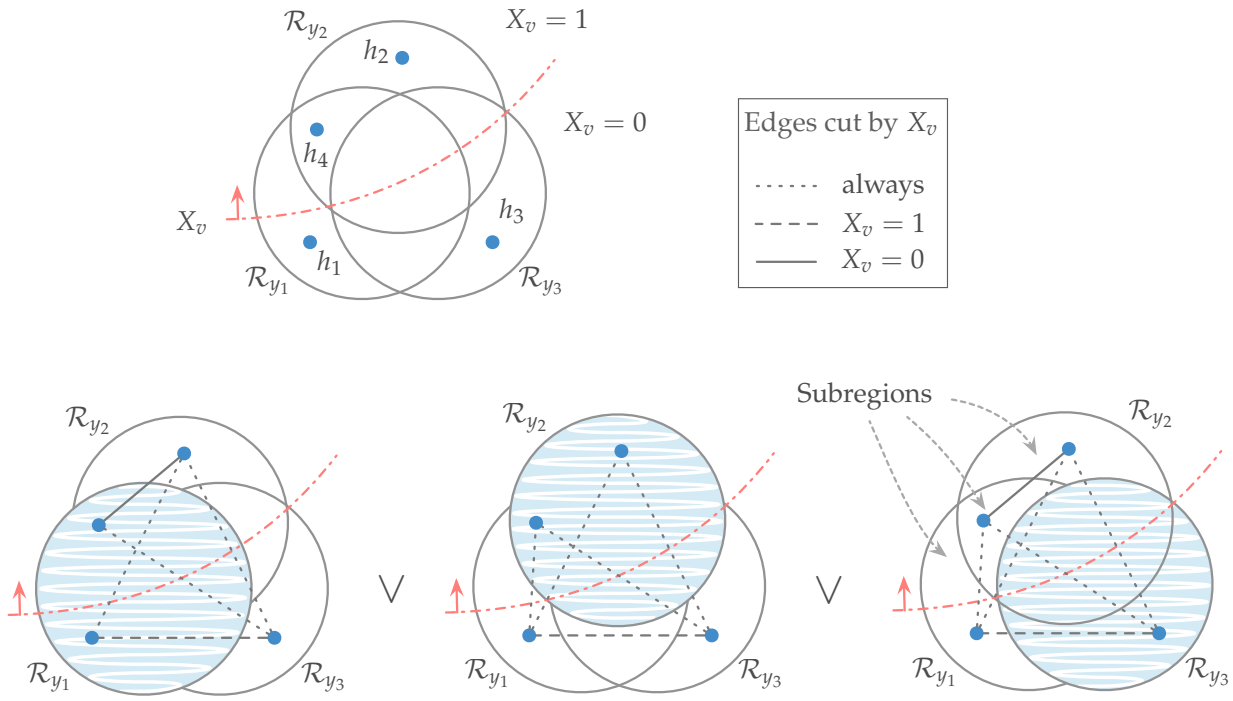


Figure 4.9: A toy DRD problem with three decision regions $\{\mathcal{R}_{y_1}, \mathcal{R}_{y_2}, \mathcal{R}_{y_3}\}$, and four possible hypotheses $\{h_1, h_2, h_3, h_4\}$. v is a test with two possible outcomes: $f_v(h_1) = f_v(h_3) = 1$ and $f_v(h_2) = f_v(h_4) = 0$. For each possible decision we can make, we construct a separate ECD problem: The three figures on the right illustrate the EC^2 graphs for each of the ECD problems. We can successfully make an optimal decision once one of the graphs is fully cut: e.g., if $X_v = 1$, the second graph is fully cut, and we identify the optimal decision y_2.

decision region from the rest. To achieve this, we model subproblem i as an ECD problem, with one of the decision regions being $\mathcal{R}_{y_i}$. Further, similarly with the construction of HEC, we partition the remaining set of hypotheses $\mathcal{H} \setminus \mathcal{R}_{y_i}$ into a collection of *subregions*, such that within each subregion, all hypotheses are contained in the same collection of decision regions from the original DRD problem. All the subregions are disjoint by definition, and hence we have a well-defined ECD problem. Solving this problem amounts to cutting all the edges between $\mathcal{R}_{y_i}$ and the subregions. See Figure 4.9 for illustration.

In this ECD problem, once all the edges are cut, either y_i is the optimal decision, or one of the subregions encodes the optimal decision. Therefore, optimizing the ECD problem associated with one of the n graphs is a *sufficient condition* for identifying the

optimal decision. Further notice that among the n ECD problems associated with the n graphs, at least one of them has to be solved (i.e., all edges cut) before we uncover the optimal decision. Therefore, we get a *necessary condition* of the DRD constraints: we have to cut all the edges in *at least one* of the n graphs. This motives us to apply a logical OR operation on the n optimization problems. Denote the EC^2 objective function for subproblem i as $f^i_{\text{EC}^2}$, and normalize them so that $f^i_{\text{EC}^2}(\emptyset) = 0$ corresponds to observing nothing and $f^i_{\text{EC}^2}(\mathbf{x}_\mathcal{V}) = 1$ corresponds to all edges being cut. We combine the objective functions $f^1_{\text{EC}^2}, \ldots, f^n_{\text{EC}^2}$ using a *Noisy-OR formulation*:

$$f_{\text{DIRECT}}(\mathbf{x}_\mathcal{A}) = 1 - \prod_i^n \left(1 - f^i_{\text{EC}^2}(\mathbf{x}_\mathcal{A})\right). \tag{4.3.1}$$

Note that by design $f_{\text{DIRECT}}(\mathbf{x}_\mathcal{A}) = 1$ iff $f^i_{\text{EC}^2}(\mathbf{x}_\mathcal{A}) = 1$ for *at least* one i. Thus, the DRD (and hence NVOI-NMU) Problem is formally equivalent to the following problem:

$$\begin{aligned} &\pi^* \in \arg\min_\pi \text{cost}(\pi), \text{ s.t.} \\ &\forall \mathbf{x}_\mathcal{V} : f_{\text{DIRECT}}(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) \geq 1 \text{ whenever } \mathbb{P}\left[\mathbf{x}_\mathcal{V}\right] > 0. \end{aligned} \tag{4.3.2}$$

The crucial advantage of this new formulation is given by the following Lemma:

Lemma 4.11. *f_{DIRECT} is strongly adaptive monotone, and adaptive submodular w.r.t. $\mathbb{P}$.*

That is, the Noisy-OR formulation for multiple EC^2 functions preserves adaptive submodularity. The proof of this result can be found in §A.2.1. These properties make f_{DIRECT} amenable for efficient greedy optimization. Formally, let $\Delta_{f_{\text{DIRECT}}}(v \mid \mathbf{x}_\mathcal{A}) := \mathbb{E}_{x_v}\left[f_{\text{DIRECT}}(\mathbf{x}_{\mathcal{A}\cup\{t\}}) - f_{\text{DIRECT}}(\mathbf{x}_\mathcal{A}) \mid \mathbf{x}_\mathcal{A}\right]$ be the expected marginal benefit in f_{DIRECT} by adding test t to $\mathbf{x}_\mathcal{A}$. The DIRECT algorithm starts with the empty set, and at each iteration, having already observed $\mathbf{x}_\mathcal{A}$, selects the test v^* with the largest benefit-to-cost ratio: $t^* \in \arg\max_v \Delta_{f_{\text{DIRECT}}}(v \mid \mathbf{x}_\mathcal{A})/c(t)$. A major benefit of adaptive submodularity is that we can use a technique called lazy evaluation to dramatically speed up the selection process [GK11a]. Further, we have the following performance guarantee:

Theorem 4.12. *Let n be the number of decisions, $p_{\min} = \min_{h\in\mathcal{H}} \mathbb{P}[h]$, π_{DIRECT} be the adaptive greedy policy w.r.t. the objective function Eq. (4.3.1). Then it holds that*

$$\text{cost}_{avg}(\pi_{\text{DIRECT}}) \leq (2n \ln(1/p_{\min}) + 1)\,\text{cost}_{avg}(\pi^*),$$

where π^* is the optimal policy for Problem (4.3.2), and hence also the NVOI-NMU and DRD Problems.

This result follows from Lemma 4.11 and Theorem 2.7. More details are given in §A.2.2. The bound of the greedy algorithm is linear in the number of decision regions. Here the factor n is a result of taking the product of n EC^2 instances. In the following, we show how this bound can often be improved.

Remark 4.13. Noisy-OR constructions have been used for classical submodular set functions [GB11b; DHK14], utilizing the fact that $f = 1 - \prod_i^n (1 - f_i)$ is submodular if each f_i is submodular. However, the function f is *not* necessarily adaptive submodular, even when each f_i is adaptive submodular and strongly adaptively monotone.

4.3.2 Improving the bound via Graph Coloring

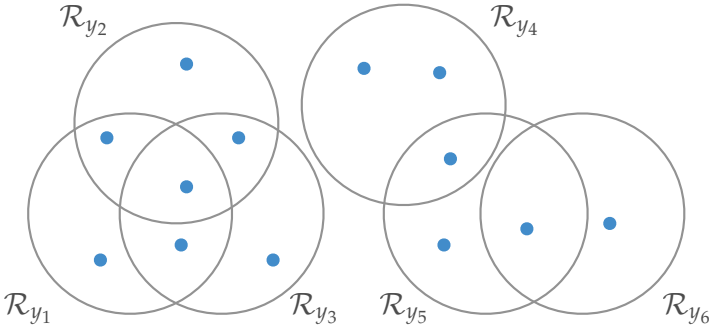
For certain applications, the number of decisions n can be large. In the extreme case where we have a unique decision for each possible observation, the bound of Theorem 4.12 becomes trivial. As noted, this is a result of taking the product of n EC^2 instances. Thus, we can improve this bound by constructing fewer instances, each with several *non-overlapping* decision regions. As long as every decision region is accounted for by at least one ECD instance, problem 4.3.2 remains equivalent to the DRD problem.

We select the sets of decision regions for each ECD instance through graph coloring. Formally, we construct an undirected graph over all decision regions, denoted as $\mathcal{G}_Y$, where we establish an edge between any pair of overlapping decision regions. That is, two decision regions $\mathcal{R}_{y_i}$ and $\mathcal{R}_{y_j}$ are adjacent in $\mathcal{G}_Y$ iff there exists a hypothesis h for which both decisions are optimal, i.e., $h \in \mathcal{R}_{y_i} \cap \mathcal{R}_{y_j}$. See Figure 4.10 for illustration. Finding a minimal set of non-overlapping decision region sets that covers all decisions is equivalent to solving a graph coloring problem, where the goal is to color the vertices of the graph $\mathcal{G}_Y$, such that no two adjacent vertices share the same color, using as few colors as possible. Thus, we can construct one ECD problem for all the decision regions of the same color, resulting in r different instances, and then use the Noisy-OR formulation to assemble these objective functions. That gives us the following theorem:

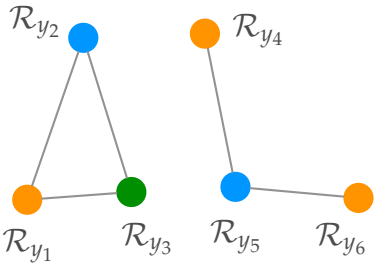
Theorem 4.14. *Let π_{DIRECT} be the adaptive greedy policy w.r.t. the objective function Eq. (4.3.1), which is computed over ECD problem instances obtained via graph coloring. Let r be the number of colors used. Then it holds that*

$$\text{cost}_{avg}(\pi_{\text{DIRECT}}) \leq (2r \ln(1/p_{\min}) + 1)\, \text{cost}_{avg}(\pi^*),$$

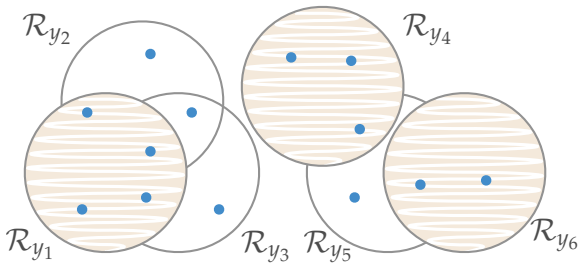
where $p_{\min} = \min_{h \in \mathcal{H}} \mathbb{P}[h]$, and π^* is the optimal policy.



(a) The DRD problem instance



(b) Coloring the DRD graph $\mathcal{G}_Y$



(c) The ECD subproblem corresponding to ●

Figure 4.10: Reducing the cost upper bound via graph coloring. We only need to construct 3 ECD subproblems to compute f_{DIRECT}, instead of 6. The middle figure shows a possible coloring assignment on the decision graph of the DRD problem. On the right, we show one example ECD problem instance, corresponding to regions $\{\mathcal{R}_{y_1}, \mathcal{R}_{y_4}, \mathcal{R}_{y_6}\}$ (colored orange). In this ECD problem instance, there are 7 disjoint regions: 3 (disjoint) decision regions $\mathcal{R}_{y_1}$, $\mathcal{R}_{y_4}$, $\mathcal{R}_{y_6}$, and 4 subregions, namely $\mathcal{R}_{y_2} \setminus (\mathcal{R}_{y_1} \cup \mathcal{R}_{y_3})$, $\mathcal{R}_{y_3} \setminus (\mathcal{R}_{y_1} \cup \mathcal{R}_{y_2})$, $(\mathcal{R}_{y_2} \cap \mathcal{R}_{y_3}) \setminus \mathcal{R}_{y_1}$, and $\mathcal{R}_{y_5} \setminus (\mathcal{R}_{y_4} \cup \mathcal{R}_{y_6})$.

While obtaining minimum graph colorings is NP-hard in general, one can show that every graph can be efficiently colored with at most one color more than the maximum vertex degree, denoted by deg, using a greedy coloring algorithm [WP67]: consider the vertices in descending order according to the degree; we assign to a vertex the smallest available color not used by its neighbors, adding a fresh color if needed. In the DRD setting, deg is the maximal number of decision regions that any decision region can be overlapped with. In practice, greedy coloring often requires far fewer colors than this upper bound. Additionally, note that when regions are disjoint, deg $= 0$ and DIRECT reverts to the EC^2 algorithm.

4.4 Efficient Optimization of VoI

The computational complexity of DIRECT depends *linearly*[1] on the number of hypotheses s in the DRD problem, i.e., the number of all possible outcome vectors $\mathbf{x}_\mathcal{V}$ in the NVOI-NMU problem. However, s can be large, in particular in settings where we model complex joint distributions $\mathbb{P}[\Theta, X_1, \ldots, X_t]$ (assuming tests have binary outcomes, s can be 2^t). Therefore it is often computationally prohibitive to keep track of all the noisy realizations of a root-cause. In this section, we focus on the computational challenge and discuss how one can exploit the structure of the probabilistic model to improve the computational efficiency of the DIRECT framework.

4.4.1 The Optimal Hypothesis Enumeration Problem

Assume that the prior $\mathbb{P}[\Theta]$ on the root-cause is known, and the prior distribution over hypotheses are fully specified by the conditional probability distribution table (CPT): $P = [\varrho_{ij}]_{t\times m}$, where $\varrho_{ij} \triangleq \mathbb{P}\left[X_v = 1 \mid \Theta = \theta_j\right]$ for $i \in [t]$ and $j \in [m]$. Let $\tilde{\mathcal{H}}$ be the set of hypotheses sampled from the CPT. Clearly, an "ideal" set $\tilde{\mathcal{H}}$ for DRD should be (1) rich enough to enclose promising candidates of *true* underlying hypotheses, and (2) compact enough so that it excludes hypotheses that are extremely rare and ensures feasibility of the algorithm. To this end, we define the *coverage* of $\tilde{\mathcal{H}}$ as its total probability mass: $Z(\tilde{\mathcal{H}}) = \sum_{h\in\tilde{\mathcal{H}}} \mathbb{P}[h]$, and the coverage of $\tilde{\mathcal{H}}$ conditioned on y as $Z(\tilde{\mathcal{H}} \mid \theta) = \sum_{h\in\tilde{\mathcal{H}}} \mathbb{P}[h \mid \Theta = \theta]$. We aim to attain a high coverage over $\mathcal{H}$ using samples while keeping the sample size as small as possible. Formally, to achieve $1-\eta$ coverage, we seek $\tilde{\mathcal{H}}^* = \arg\min_{\tilde{\mathcal{H}}: Z(\tilde{\mathcal{H}})\geq 1-\eta} |\tilde{\mathcal{H}}|$.

Existing approaches for generating hypotheses, such as Monte-Carlo sampling, often require a very large sample size to reach a certain coverage of the total probability mass. To illustrate this, let us consider a simple multinomial distribution that describes the probability distribution of four mutually exclusive hypotheses (h_1, h_2, h_3, h_4), with probabilities (0.94, 0.03, 0.02, 0.01). Suppose that we have a hypothesis generator that directly samples hypotheses according to their probabilities (as we were rolling a dice). If we require observing a subset of hypotheses that cover at least 98% of the total mass

[1] Since DIRECT requires the computation of r EC^2 scores, the computational complexity of DIRECT is *linear* in both r (the number of colors used) and s (the number of hypotheses).

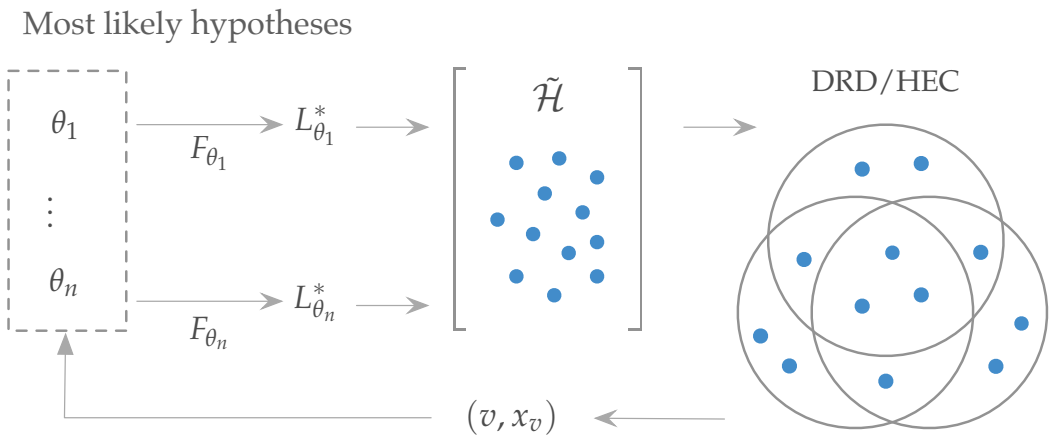


Figure 4.11: The dynamic hypothesis enumeration framework.

(i.e. h_1, h_2 and at least one of h_3 or h_4) with a confidence level of at least 99%, then we need at least a sample of average size 174, to cover the "rare" observations.

4.4.2 Dynamic Hypothesis Enumeration

The problem of the Monte-Carlo approach is that it lacks consideration of the structure of the VoI problem. Instead, our method aims at providing the most likely configurations – covering up to a pre-specified fraction of the total probability mass – in an efficient and adaptive way. In a nutshell, we adaptively maintain a pool of hypotheses that constitute a sample with small size and coverage.

In particular, our sampling scheme consists of two modules: (1) Algorithm 2 *locally* enumerates the *most likely* hypotheses for each hidden state, which will cover – by taking the union over all hidden states – at least $(1 - \eta)$ fraction of the total probability mass of all hypotheses; and (2) Algorithm 3 provides a *global* mechanism that, after observing a test outcome, adaptively filters out inconsistent hypotheses and *re-samples* new hypotheses; re-sampling is done by calling Algorithm 2, to ensure that the new sample's coverage is sufficient to derive reliable statistics when deriving the new optimal test to be performed. The overall framework is illustrated in Fig. 4.11.

Generating Hypotheses for Each Hidden State

The basic module of our hypothesis enumeration framework is a "local" hypothesis generator, which enumerates the most likely hypotheses for any given hidden state. It incrementally builds a Directed Acyclic Graph (DAG) of hypotheses, starting from the most likely configuration. At each step, the leaf nodes of the DAG represent the current *candidate frontier*, i.e., the set of hypotheses that dominate all other candidate hypotheses in terms of likelihood. This set is used to generate the remaining hypotheses through a "children generation" mechanism: the next most likely hypothesis of *candidate frontier* is identified, and its (at most two) children are added as new leaf nodes to the DAG.

The input of Algorithm 2 consists of the given hidden state value θ, the associated outcome probability vector over n tests, i.e., $\mathbb{P}[x_v \mid \theta]$ $(v = 1, \ldots, t)$, and the threshold of coverage η. Optionally, it might be given a candidate frontier F_θ, which is defined as a list of consistent hypotheses h with their log-probability weights $\lambda_\theta(h) = \mathbb{P}[h \mid \theta, \mathbf{x}_\mathcal{A}]$. F_θ is obtained as a by-product when calling the same module for the same θ at the previous iterations and is used as a seed set of nodes to further expand the DAG.

W.l.o.g., we can assume that tests' outcomes are defined in such a way that $\mathbb{P}[X_v = 1 \mid \theta] \geq 0.5$ (if not, we can redefine a test so that the label is flipped). Initially (line 2), the tests are rearranged in decreasing order of $\mathbb{P}[X_v = 1 \mid \theta]$. Thereby, the last test will be the one with the highest uncertainty; hence flipping the sign of this test will have the minimal effect on the overall likelihood. The generator then proceeds to enumerate the most likely hypotheses corresponding to the given hidden state θ. At line 9, the two children hypotheses are generated as follows. For the first child, if the last (right-most) bit of h^* is 1, we then create h_{c_1} by switching the last bit to 0. For instance, the child hypothesis h_{c_1} of $h^* = [0,1,1,0,1]$ is $[0,1,1,0,0]$. Its log-probability is obtained by $\lambda_\theta(h_{c_1}) = \lambda_\theta(h^*) + q_n - p_n$. For the second child, we first need to locate the right-most "$[1,0]$" pair in h^* (if there exists any; otherwise we do nothing), and the create h_{c_2} by switching "$[1,0]$" into "$[0,1]$". For instance, the child hypothesis h_{c_2} of $h^* = [0,1,1,0,1]$ is $[0,1,0,1,1]$. Its associated log-probability is computed by $\lambda_\theta(h_{c_2}) = \lambda_\theta(h^*) + q_v - p_v + p_{v+1} - q_{v+1}$, where i is the bit index of the "1" in the right-most "$[1,0]$" pair.

As output, Algorithm 2 produces a ranked list L^*_θ of the most likely hypotheses for a given θ, and their log-probabilities $\lambda_\theta(h) = \log(\mathbb{P}[h \mid \theta, \mathbf{x}_\mathcal{A}])$, such that $\sum_{h \in L^*_\theta} \exp(\lambda_\theta(h)) \geq (1 - \eta)$. In addition, it also produces a residual frontier F_θ that will be used, after

Algorithm 2: Generate the most likely hypotheses for root-cause θ

Input: Root-cause θ, Conditional probability table P, coverage threshold η, (optional) frontier F_θ;
begin
  Sort tests in decreasing order of $\mathbb{P}[X_v = 1 \mid \theta]$;
  foreach $v \in \{1, \ldots, t\}$ **do**
    $p_v \leftarrow \log(\mathbb{P}[X_v = 1 \mid \theta])$;
    $q_v \leftarrow \log(\mathbb{P}[X_v = 0 \mid \theta])$;
  end
  if F_θ *is empty* **then**
    $F_\theta \leftarrow \{h_1 = [1, 1, \ldots, 1]\}$, with log-weight $\lambda_\theta(h_1) = \sum_v \log p_v$;
    $L^*_\theta \leftarrow \emptyset$;
  end
  while $\sum_{h \in L^*_\theta} \exp(\lambda_\theta(h)) < (1 - \eta)$ **do**
    $h^* \leftarrow \arg\max_{h \in F_\theta} \lambda_\theta(h)$;
    $F_\theta \leftarrow F_\theta \setminus \{h^*\}$, $L^*_\theta \leftarrow L^*_\theta \cup \{h^*\}$;
    Generate (at most) 2 children h_{c_1}, h_{c_2} from h^*;
    $F_\theta \leftarrow F_\theta \cup \{h_{c_1}, h_{c_2}\}$;
  end
  Output: Most likely hypotheses L^*_θ for θ, log- probabilities $\lambda_\theta(h) = \log(\mathbb{P}[h \mid \theta, \mathbf{x}_\mathcal{A}])$, and F_θ.
end

filtering and transformation, as a new "seed" list for next iteration.

Iterative Filtering and Hypothesis Re-sampling

After generating the most likely hypotheses for each hidden state, we merge them into a global set and compute their marginal likelihoods. We dynamically re-sample new hypotheses as more observations are made. Re-sampling is necessary to constantly guarantee that the sample set that covers at least $1 - \eta$ of the total remaining mass, after new observations (test outcomes) become available. The detailed description of the module is provided in Algorithm 3.

Algorithm 3: Iterative Filtering and Re-sampling

Input: Conditional probability table $\boldsymbol{\theta}$, Prior $\mathbb{P}[\Theta]$, coverage threshold η;
begin
  $\tilde{\mathcal{H}} \leftarrow \emptyset$;
  while *stopping condition for* EC^2 *not reached* **do**
    foreach $y \in \{y_1, \ldots, y_m\}$ **do**
      Call Algorithm 2 to generate L^*_θ;
      $\tilde{\mathcal{H}} \leftarrow \tilde{\mathcal{H}} \cup L^*$;
    end
    foreach $h \in \tilde{H}$ **do**
      $p(h \mid \mathbf{x}_\mathcal{A}) \leftarrow \sum_\theta \exp(\lambda_\theta(h)) \cdot \mathbb{P}[\theta \mid \mathbf{x}_\mathcal{A}]$;
    end
    Run DiRECt to determine the next test v; $\mathcal{A} \leftarrow \mathcal{A} \cup \{v\}$;
    Observe x_v; $\mathbf{x}_\mathcal{A} \leftarrow \mathbf{x}_\mathcal{A} \cup \{x_v\}$;
    Update $\mathbb{P}[\theta \mid \mathbf{x}_\mathcal{A}]$;
    $\lambda_\theta(h) \leftarrow \lambda_\theta(h) - \log \mathbb{P}[x_v \mid \theta]$;
    Filter out inconsistent hypotheses in L^*_θ and F_θ;
    Remove test v from the list of available tests;
  end
  Output: (test - outcome) vectors $\mathbf{x}_\mathcal{A}$, decision R
end

The global iterative filtering and re-sampling module consists of a global loop, where after initializing all ranked lists L^*_θ to $\emptyset$ and $\mathbb{P}[\theta \mid \mathbf{x}_\mathcal{A} = \emptyset]$ to the prior distribution over the hidden states, it iteratively performs the following sequences of operations: First, for each hidden state θ, it calls Algorithm 2 to generate enough hypotheses so that L^*_θ covers at least $(1-\eta)$ of its current mass, i.e., $Z(L^*_\theta \mid \theta, \mathbf{x}_\mathcal{A}) \geq 1-\eta$ (line 3). L^*_θ might not be initially empty due to a previous call to Algorithm 2. In this case, the generator produces only new additional hypotheses starting from the frontier F_θ until the desired coverage is achieved. This step is not necessary for the θ's that are inconsistent with $\mathbf{x}_\mathcal{A}$, i.e., for those hidden states whose posterior distribution given $\mathbf{x}_\mathcal{A}$ is zero.

By construction, once we merge the hypotheses associated with each hidden state (line 4), the sample set $\tilde{\mathcal{H}}$ covers at least $(1-\eta)$ fraction of the total mass that

is consistent with all the observations up to $\mathbf{x}_{\mathcal{A}}$: $Z(\tilde{\mathcal{H}} \mid \mathbf{x}_{\mathcal{A}}) = \sum_{h \in \tilde{\mathcal{H}}} \mathbb{P}[h \mid \mathbf{x}_{\mathcal{A}}] \geq \sum_{\theta} \sum_{h \in L_{\theta}^{*}} \mathbb{P}[h \mid \theta, \mathbf{x}_{\mathcal{A}}] \mathbb{P}[\theta \mid \mathbf{x}_{\mathcal{A}}] \geq \sum_{\theta} (1-\eta) \mathbb{P}[\theta \mid \mathbf{x}_{\mathcal{A}}] = 1-\eta$. The procedure is then followed by performing EC^2 (or any other submodular surrogates-based greedy algorithm) on $\tilde{\mathcal{H}}$ to identify the next test to be performed (see Fig. 4.11).

4.4.3 Theoretical Analysis

With initial Samples Only. Assume that we only sample the hypotheses *once* at the beginning of each experiment, i.e., we do not resample the hypotheses after observing the outcome of a test. If the underlying true hypothesis is included in the sampled set $\tilde{\mathcal{H}}$, then by construction Algorithm 3 is guaranteed to make the optimal decision. Otherwise, with small probability it fails to output the optimal decision. Theorem 4.15 states a tradeoff between the size of $\tilde{\mathcal{H}}$ and the expected cost of Algorithm 3.

Theorem 4.15. *Suppose we have generated hypotheses $\tilde{\mathcal{H}}$ with coverage $1-\eta$. Define $\tilde{p}_{\min} = \min_{h \in \tilde{\mathcal{H}}} \frac{\mathbb{P}[h]}{1-\eta}$. Let $\pi^{g}_{\tilde{\mathcal{H}}}$ be the policy induced by Algorithm 3, $\pi^{*}_{\mathcal{H}}$ be the optimal policy on the original distribution of $\mathcal{H}$, and $c(\mathcal{T})$ be the cost of performing all tests. Then, it holds that*

$$\mathrm{cost}_{avg}\left(\pi^{g}_{\tilde{\mathcal{H}}}\right) \leq \left(r \ln\left(1/\tilde{p}_{\min}\right) + 1\right) \mathrm{cost}_{avg}(\pi^{*}_{\mathcal{H}}) + \eta \cdot c(\mathcal{T}).$$

Moreover, if we stop running $\pi^{g}_{\tilde{\mathcal{H}}}$ once it cuts all edges on $\tilde{\mathcal{H}}$, then with probability at least $1-\eta$, $\pi^{g}_{\tilde{\mathcal{H}}}$ outputs the optimal decision with

$$\mathrm{cost}_{wc}\left(\pi^{g}_{\tilde{\mathcal{H}}}\right) \leq \left(r \ln(1/\tilde{p}_{\min}) + 1\right) \mathrm{cost}_{wc}(\pi^{*}_{\mathcal{H}}).$$

Hereby, r denotes the number of colors involved in the DiRECt *subroutine of Algorithm 3.*

We defer the proof to §A.2.3. Note that the expected cost is computed w.r.t. the original hypothesis distribution $\mathbb{P}[H \mid H \in \mathcal{H}]$. Theorem 4.15 establishes a bound between the cost of the greedy algorithm on the samples $\tilde{\mathcal{H}}$, and the cost of the optimal algorithm on the total population $\mathcal{H}$. The quality of the bound depends on η, as well as the structure of the problem (which determines $\tilde{p}_{\min}$). Running the greedy policy on a larger set of samples leads to a lower failure rate, although $\tilde{p}_{\min}$ might be significantly smaller for small η. Further, with adaptive re-sampling we constantly maintain a $1-\eta$ coverage on the posterior distribution over $\mathcal{H}$. With similar reasoning, we can show that the greedy policy with adaptively-resampled posteriors yields a lower failure rate than the greedy policy which only samples the hypotheses once at the start of the session.

With Resampling. We now consider the policy which greedily picks tests according to the adaptively-sampled posterior distribution of H at each iteration.

Theorem 4.16. *Let k, ℓ be positive integers[2], f be the EC^2 objective function, $\pi_{\tilde{\mathcal{H}},[\ell]}$ be the greedy policy on $\tilde{\mathcal{H}}$, and $\pi^*_{\mathcal{H},[k]}$ be the optimal policy that achieves the maximal expected utility under budget k on $\mathcal{H}$, and*

$$F\left(\pi_{\tilde{\mathcal{H}},[\ell]}\right) \geq \left(1 - e^{-\ell/k}\right) F\left(\pi^*_{\mathcal{H},[k]}\right) - k\epsilon$$

where $\epsilon = 2\eta\left(1 - \left(\frac{1}{k}\right)^{\ell}\right)$, and $F(\pi) \triangleq \mathbb{E}_{h\sim\mathbb{P}[\mathcal{H}]}[f(\mathcal{S}(\pi, h))]$ denotes the expected utility of running policy π w.r.t. the original distribution.

Note that the above result applies to the EC^2 algorithm with *adaptive-resampled* posteriors at each iteration. The additive term $k\epsilon$ on the RHS is due to the incompleteness of the samples provided by the sampling algorithm. The main intuition behind the proof is that, due to the effect of resampling, the expected one-step gain of the greedy policy $\pi_{\tilde{\mathcal{H}},[\ell]}$ on the sampled distribution suffers a small loss at each iteration, comparing to the greedy algorithm on the true distribution. The loss will be accumulated after ℓ rounds, leading to a cumulative loss of up to $k\epsilon$ in the lower bound.

We defer the proof of Theorem 4.16 to Section A.2.4. In the following, we show that an additive term is necessary for the lower bound (i.e., we cannot remove the additive term in the bound, and push it into the multiplicative term involving $1 - e^{-\ell/k}$).

Suppose there are two root-causes θ_1, θ_2 and two tests with outcomes denoted by X_1 and X_2. Let $\eta = 0.1$. The conditional probabilities for the test outcomes are as follows: $\mathbb{P}[X_1 = 1 \mid \theta_1] = \mathbb{P}[X_1 = 1 \mid \theta_2] = 1, \mathbb{P}[X_2 = 1 \mid \theta_1] = 0.001, \mathbb{P}[X_2 = 1 \mid \theta_2] = 0$. There are only two hypotheses with non-zero probability, i.e., $h_1 = [1, 0]$ and $h_2 = [1, 1]$. Under some utility assignment, h_1 and h_2 will be assigned to different decision regions.

However, the hypothesis enumeration algorithm will output only one hypothesis $h_1 = [1, 0]$, since $\mathbb{P}[h_1 \mid \theta_1] > 1 - \eta$ and $\mathbb{P}[h_2 \mid \theta_2] > 1 - \eta$. Assume that we further add infinitely many "dummy tests", i.e., for all v in this set, $p(X_v = 1 \mid \theta) = 0$ for all θ. Then the greedy algorithm will choose those tests with high probability, since the gain for all tests over $\tilde{\mathcal{H}}$ is 0; whereas a smarter algorithm will pick test X_2, because we can identify the target region (and hence obtain a positive gain) upon observing its outcome.

[2]If we assume unit cost for all tests, then k, ℓ are the number of tests selected. Otherwise, with non-uniform test costs, k, ℓ are the budget on the cost of selected items.

4.5 ECED: Optimizing VoI in the Presence of Noise

In principle, we can convert any NVOI-NMU problem (Problem 4.1.2) into a DRD problem (Problem 4.1.3) by enumerating hypotheses from the prior $\mathbb{P}[\Theta, X_1, \ldots, X_t]$, and solve it near-optimally under the submodular surrogate-based framework (e.g., by running DIRECT or HEC). Theorem 4.14 provides an upper bound on the cost that π_{DIRECT} takes to solve the DRD problem - which requires it to cut *all* the edges (in at least one ECD problem instance) - against the optimal policy that does so. In practice, due to the effect of noise, we may have to exhaust all the tests before we can cut all the edges. Being "absolutely certain" about the target decision is clearly an unrealistic goal for most applications. A natural question to ask is, how can we design an algorithm which attains near-optimal performance, if we are not required to identify the optimal decision with absolute certainty, but just to solve the DRD problem with high confidence?

As discussed in §4.4, one way to address this problem is to run DIRECT on the sampled hypotheses instead of the original distribution, which may lead to an early stop. Our theoretical result (Theorem 4.15) bounds the cost of the sample-based algorithm against the optimal algorithm that *solves* the DRD problem. However, a stronger baseline we wish to compare to is the optimal policy that also solves the DRD with *high confidence*. In this section, we investigate an approach alternative to the sampling-based method and prove strong theoretical guarantees of this flavor.

4.5.1 The Noisy ECD Problem

For the sake of theoretical analysis, in this subsection, we focus on the *equivalence class determination* (ECD) setting, where decision regions do not overlap. Concretely, we assume that there are m possible root-causes and n possible decisions, and the value of target decision Y depends on the root-cause Θ through a deterministically mapping $r : \{\theta_1, \ldots, \theta_m\} \to \{y_1, \ldots, y_n\}$ that gives $Y = r(\Theta)$. We further assume that tests have *binary* outcomes, and each test incurs a *unit* cost.

To simplify notation, we use $\psi_\pi := \mathcal{S}(\pi, \mathbf{x}_\mathcal{V}) \in 2^{\mathcal{V}\times\mathcal{X}}$ to denote a path seen by policy π as is similarly done in §3.3.2 (recall its graphical representation in Fig. 3.4). We use $\psi_\ell := \mathcal{S}(\pi_{[\ell]}, \mathbf{x}_\mathcal{V}) \in 2^{\mathcal{V}\times\mathcal{X}}$ to denote a path seen by policy π if it has selected ℓ tests. Once ψ_π is observed, we obtain a new posterior on Θ (and consequently on Y). The

best prediction one can thus make under the Bayesian setting is the MAP estimator $\hat{y}$ of Y, i.e., $\hat{y} \triangleq \arg\max_{y' \in \mathcal{Y}} \mathbb{P}[Y = y' \mid \psi_\pi]$. The error probability of predicting $\hat{y}$ is

$$p_{\text{ERR}}^{\text{MAP}}(\psi_\pi) \triangleq \mathbb{P}[\hat{y} \neq y \mid \psi_\pi] = 1 - \max_{y \in \mathcal{Y}} \mathbb{P}[y \mid \psi_\pi].$$

We call $p_{\text{ERR}}^{\text{MAP}}$ the *prediction error* of the MAP estimator. The expected prediction error after running policy π is then defined as $p_{\text{ERR}}(\pi) \triangleq \mathbb{E}_{\psi_\pi}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\pi)\right]$.

Given some small tolerance $\delta \in [0, 1]$, we seek a policy with the minimal cost, such that upon termination, the posterior puts at least $1 - \delta$ mass on the most likely target value y in expectation. In other words, we require that the expected prediction error after running the policy is at most δ. Denote such policy by $\text{OPT}(\delta)$. Formally, we seek

$$\text{OPT}(\delta) \in \arg\min_{\pi} \text{cost}(\pi), \text{ s.t. } p_{\text{ERR}}(\pi) \leq \delta. \tag{4.5.1}$$

When $\delta = 0$, Problem 4.5.1 reduces to the DRD problem (Problem 4.1.3). Therefore it is a strict generalization of the later. In this section, we consider bounding the worst-case cost, which is defined as $\text{cost}_{wc}(\pi) \triangleq \max_{\psi_\pi} |\psi_\pi|$, i.e., the maximum number of tests performed by π over all possible paths it takes.

Remark 4.17. A natural approach to solving Problem 4.5.1 for $\delta \geq 0$ would be to pick tests greedily maximizing the expected reduction in the error probability p_{ERR}. This, in fact, in the myopic VoI policy, if we define the utility of making decision y on root-cause θ to be the $1/0$ loss function, i.e., $u(\theta, y) = \mathbb{1}\{r(\theta) = y\}$. As we have shown in §4.2.1, posterior-based approaches (including this one) can perform arbitrarily badly for the decision-making problem we consider. Therefore, we aim to optimize a surrogate objective function which captures the effect of noise, while being amenable to greedy optimization.

Remark 4.18. Note that there are different ways of defining "success" of a policy. Other than bounding the prediction error as considered in Eq. (4.5.1), an alternative option is to ensure that the *excess error* or *regret* of acting upon ψ_π, compared to having observed all the tests is not more than δ. While the regret-based success criterion might be an sensible alternative criterion to consider, the prediction error criterion offers a natural stopping condition for running a policy (as one can compute the $p_{\text{ERR}}^{\text{MAP}}(\psi_\pi)$ purely based on the posterior). Hence we focus on Problem 4.5.1 throughout this section.

4.5.2 The ECED Algorithm

We now introduce ECED, a principled algorithm for solving the noisy ECD problem. Different from the DIRECT and its "noisy" extension (§4.3, §4.4), ECED no longer needs to explicitly enumerate *hypotheses* (i.e., possible outcomes of all tests). Instead, it directly draws edges between root-causes that do not share a region, i.e., $\mathcal{E} \triangleq \{(\theta, \theta') : r(\theta) \neq r(\theta')\}$, and scores tests based on the weight of those edges.

ECED with Bayesian Updates on Edge Weights In general, test outcomes are not necessarily deterministic given a root-cause, i.e., $\forall \theta,\ \mathbb{P}[X_v \mid \theta] \in [0,1]$. One can not "cut away" a root-cause θ by observing x_v, as long as $\mathbb{P}[X_v = x_v \mid \theta] > 0$. In such cases, a natural extension of the edge-cutting strategy will be – instead of cutting off edges – to *discount* the edge weights through Bayesian updates: After observing x_v, we can discount the weight of an edge (θ, θ'), by multiplying the probabilities of its incident root-causes with the likelihoods of the observation: $w((\theta, \theta') \mid x_v) := \mathbb{P}[\theta]\,\mathbb{P}[\theta'] \cdot \mathbb{P}[x_v \mid \theta]\,\mathbb{P}[x_v \mid \theta'] = \mathbb{P}[\theta, x_v] \cdot \mathbb{P}[\theta', x_v]$. Here we choose *not* to normalize the probabilities of θ, θ' to their posterior probabilities. Otherwise, we can end up having 0 gain in terms of edge weight reduction, even if we perform a very informative test. This gives us a greedy policy that, at every iteration, picks the test that has the maximal expected reduction in total edge weight. We call such policy EC^2-Bayes.

Unfortunately, as we demonstrate later in §4.6, this seemingly promising update scheme is not ideal for solving our problem: it tends to pick tests that are very noisy, which do not help facilitate differentiation among different target values. Consider a simple example as illustrated in Fig. 4.12. There are three root-causes distributed as $\mathbb{P}[\theta_1] = 0.2, \mathbb{P}[\theta_2] = \mathbb{P}[\theta_3] = 0.4$, and two target values $r(\theta_1) = r(\theta_2) = y_1, r(\theta_3) = y_2$. We want to evaluate two tests: (1) a purely noisy test v_1, i.e., $\forall \theta,\ \mathbb{P}[X_{v_1} = 1 \mid \theta] = 0.5$, and (2) a noiseless test v_2 with $\mathbb{P}[X_{v_2} = 1 \mid \theta_1] = 1$ and $\mathbb{P}[X_{v_2} = 1 \mid \theta_2] = \mathbb{P}[X_{v_2} = 1 \mid \theta_3] = 0$. One can easily verify that by running EC^2-Bayes, one actually prefers v_1 (with expected reduction in edge weight 0.18, as opposed to 0.112 for v_2).

The ECED Algorithm The example above hints us on an important principle of designing proper objective functions for this task: as the noise rate increases, one must take reasonable precautions when evaluating the informativeness of a test, such that the undesired contribution by noise is accounted for. Suppose we have performed test

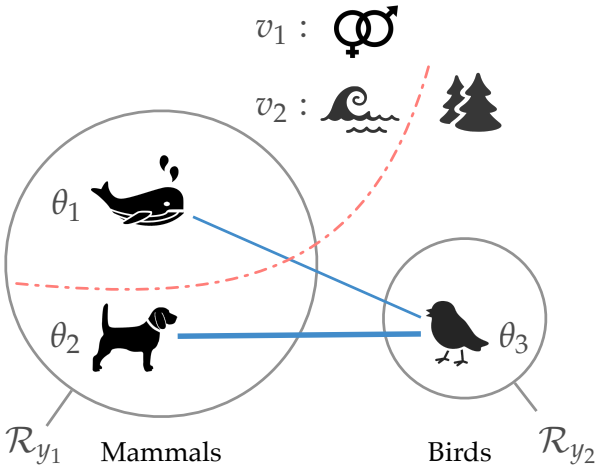


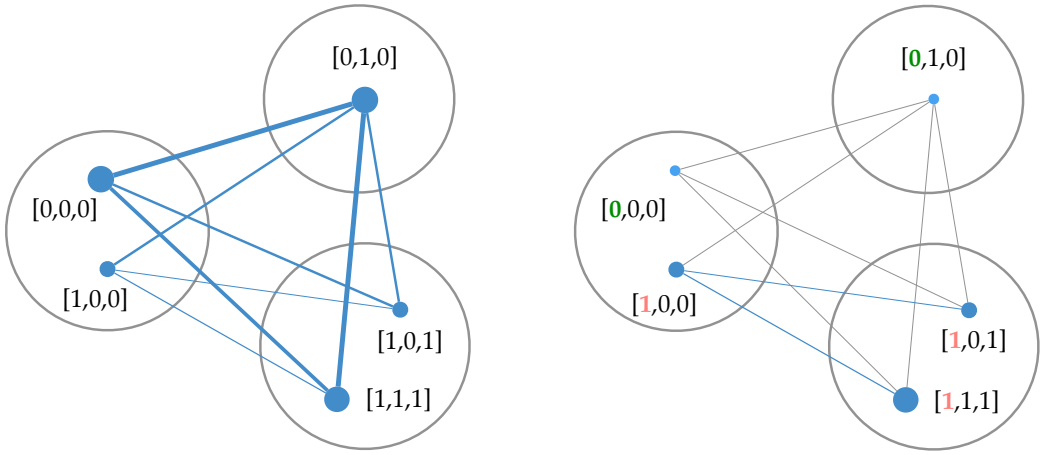
Figure 4.12: An illustrative example for EC^2-Bayes and ECED. There are two tests, v_1 is very informative, as observing its outcome may immediately tell us which region is correct (e.g., if X_{v_2} = "aquatic animals", then we know the target is "mammal"). v_2, on the other hand, can be viewed as a "purely noisy" test, because knowing the gender doesn't change our belief on the root-causes. Hence, we want to design a criterion that encourages picking v_2.

v and observed x_v. We call a root-cause θ to be "consistent" with observation x_v, if x_v is the most likely outcome of X_v given θ (i.e., $x_v \in \arg\max_x \mathbb{P}[X_v = x \mid \theta]$). Otherwise, we say θ is inconsistent. Now, instead of discounting the weight of all root-causes by the likelihoods $\mathbb{P}[X_v = x_v \mid \theta]$ (as EC^2-Bayes does), we choose to discount the root-causes by the *likelihood ratio*:

$$\lambda_{\theta,x_v} \triangleq \frac{\mathbb{P}[X_v = x_v \mid \theta]}{\max_{x'_v} \mathbb{P}[X_v = x'_v \mid \theta]}.$$

Intuitively, this is because we want to "penalize" a root-cause (and hence the weight of its incident edges), only if it is *inconsistent* with the observation (See Fig. 4.13). When x_v is consistent with root-cause θ, then $\lambda_{\theta,x_v} = 1$ and we do not discount θ; otherwise, if x_v is inconsistent with θ, we have $\lambda_{\theta,x_v} < 1$. When a test is not informative for root-cause θ, i.e. $\mathbb{P}[X_v \mid \theta]$ is uniform, then $\lambda_{\theta,e} = 1$, so that it neutralizes the effect of such test in terms of edge weight reduction.

Formally, given observations ψ_π, we define the (basic) value of observing x_v as the total



(a) Initializing the ECD graph

(b) Observing $X_1 = 1$.

Figure 4.13: Illustration of the equivalence class edge discounting algorithm. Hypotheses are represented in dots. The size of a dot is proportional to its probabilities. Upon observing "inconsistent" outcomes, we discount the hypothesis accordingly and consequently discount its incident edges.

amount of edge weight discounted:

$$\delta_{\text{BS}}(x_v \mid \psi_\pi) \triangleq \sum_{(\theta,\theta')\in\mathcal{E}} \mathbb{P}\left[\theta, \psi_\pi\right] \mathbb{P}\left[\theta', \psi_\pi\right] \cdot (1 - \lambda_{\theta,x_v}\lambda_{\theta',x_v}).$$

Further, we call test v to be *non-informative*, if its outcome does not affect the distribution of Θ, i.e., $\forall\, \theta, \theta' \in \text{supp}(\Theta)$ and $x_v \in \mathcal{X}$, $\mathbb{P}\left[X_v = x_v \mid \theta\right] = \mathbb{P}\left[X_v = x_v \mid \theta'\right]$. Obviously, performing a non-informative test does not reveal any useful information of Θ (and hence Y). Therefore, we should augment our basic value function δ_{BS}, such that the value of a non-informative test is 0. Following this principle, we define

$$\delta_{\text{OFFSET}}(x_v \mid \psi_\pi) \triangleq \sum_{(\theta,\theta')\in\mathcal{E}} \mathbb{P}\left[\theta, \psi_\pi\right] \mathbb{P}\left[\theta', \psi_\pi\right] \cdot (1 - \max_\theta \lambda^2_{\theta,x_v}),$$

as the *offset* value for observing outcome x_v. It is easy to check that if test v is non-informative, then it holds that $\delta_{\text{BS}}(x_v \mid \psi_\pi) - \delta_{\text{OFFSET}}(x_v \mid \psi_\pi) = 0$ for all $x_v \in \mathcal{X}$; otherwise $\delta_{\text{BS}}(x_v \mid \psi_\pi) - \delta_{\text{OFFSET}}(x_v \mid \psi_\pi) \geq 0$. This motivates us to use the following objective function:

$$\Delta_{\text{ECED}}(v \mid \psi_\pi) \triangleq \mathbb{E}_{x_v}[\delta_{\text{BS}}(x_v \mid \psi_\pi) - \delta_{\text{OFFSET}}(x_v \mid \psi_\pi)]\,, \tag{4.5.2}$$

Algorithm 4: The Equivalence Class Edge Discounting (ECED) Algorithm

Input: $[\lambda_{\theta,x}]_{m\times t}$ (or Conditional Probabilities $\mathbb{P}[X \mid \Theta]$), Prior $\mathbb{P}[\Theta]$, Mapping $r : \mathrm{supp}(\Theta) \to \mathcal{Y}$;
begin
  $\psi_\pi \leftarrow \emptyset$;
  foreach $(\theta, \theta') \in \mathcal{E}$ **do**
    $w_{\theta,\theta'} \leftarrow \mathbb{P}[\theta]\,\mathbb{P}[\theta']$;
  end
  while $p_{\mathrm{ERR}}^{\mathrm{MAP}}(\psi_\pi) > \delta$ **do**
    $v^* \leftarrow \arg\max_v \mathbb{E}_{x_v}\Big[\sum_{(\theta,\theta')\in\mathcal{E}} w_{\theta,\theta'} \cdot \big(\overbrace{1-\lambda_{\theta,x_v}\lambda_{\theta',x_v}}^{\text{weight discounted}} - \overbrace{(1-\max_{\theta''}\lambda^2_{\theta'',x_v})}^{\text{offset term}}\big)\Big]$;
    Observe x_{v^*}; $w_{\theta,\theta'} \leftarrow w_{\theta,\theta'} \cdot \mathbb{P}[x_{v^*} \mid \theta]\,\mathbb{P}[x_{v^*} \mid \theta']$;
    $\psi_\pi \leftarrow \psi_\pi \cup \{(v^*, x_{v^*})\}$;
  end
  Output: $y^* = \arg\max_y \mathbb{P}[y \mid \psi_\pi]$.
end

as the expected amount of edge weight that is effectively reduced by performing test v. We call the algorithm that greedily maximizes Δ_{ECED} the *Equivalence Class Edge Discounting* (ECED) algorithm and present the pseudocode in Algorithm 4.

4.5.3 Theoretical Analysis

Upper bounds on the worst-case cost Similar with EC^2, both the *computation complexity* (i.e., the running time) and the *query complexity* (i.e., the number of tests needed) of ECED depends on the number of root-causes. Let $\epsilon_{\theta,v} \triangleq 1 - \max_x \mathbb{P}[X_v = x \mid \theta]$ be the noise rate for test v. As our main theoretical result, we show that under the basic setting where test outcomes are *binary*, and the test noise is *independent* of the underlying root-causes (i.e., $\forall\theta \in \mathrm{supp}(\Theta)$, $\epsilon_{\theta,v} \equiv \epsilon_v$), ECED is competitive with the optimal policy that achieves a lower error probability for Problem (4.1.3):

Theorem 4.19. *Fix $\delta \in (0,1)$. To achieve expected error probability less than δ, it suffices to run ECED for $O\left(\frac{k}{c_\varepsilon}\left(\log\frac{km}{\delta}\log\frac{m}{\delta}\right)^2\right)$ steps where $m \triangleq |\mathrm{supp}(\Theta)|$ denotes the number of root-causes, $c_\varepsilon \triangleq \min_{v\in\mathcal{V}}(1-2\epsilon_v)^2$ characterizes the severity of noise, and $k \triangleq$*

$\text{cost}_{wc}\left(\text{OPT}(\delta_{\text{OPT}})\right)$ *is the worst-case cost of the optimal policy that achieves expected error probability* $\delta_{\text{OPT}} \triangleq O\left(\frac{\delta}{(\log m \cdot \log(1/\delta))^2}\right)$.

Note that a pessimistic upper bound for k is the total number of tests t, and hence the cost of ECED is at most $O\left((\log(tm/\delta)\log(m/\delta))^2/c_\varepsilon\right)$ times the worst-case cost of the optimal algorithm, which achieves a lower error probability $O\left(\delta/(\log m \cdot \log(1/\delta))^2\right)$. Further, as one can observe, the upper bound on the cost of ECED degrades as we increase the maximal noise rate of the tests. When $c_\varepsilon = 1$, we have $\epsilon_v = 0$ for all test v, and ECED reduces to the EC^2 algorithm. Theorem 4.19 implies that running EC^2 for $O\left(k\left(\log\frac{km}{\delta}\log\frac{m}{\delta}\right)^2\right)$ in the noise-free setting is sufficient to achieve $p_{\text{ERR}} \leq \delta$. Finally, notice that by construction ECED never selects any non-informative test. Therefore, we can always remove purely noisy tests (i.e., $\{v : \forall\theta,\ \mathbb{P}[X_v = 1 \mid \theta] = \mathbb{P}[X_v = 0 \mid \theta] = 1/2\}$), so that $c_\varepsilon > 0$, and the upper bound in Theorem 4.19 becomes non-trivial.

Information-theoretic Auxiliary Function We now present the main idea behind the proof of Theorem 4.19. In general, an effective way to relate the performance (measured in terms of the gain in the target objective function) of the greedy policy to the optimal policy is by showing that, the *one-step* gain of the greedy policy always makes effective progress towards approaching the cumulative gain of OPT *over k steps*. One powerful tool facilitating this is the *adaptive submodularity* theory, which imposes a lower bound on the one-step greedy gain against the optimal policy, given that the objective function in consideration exhibits a natural diminishing returns condition. Unfortunately, in our context, the target function to optimize, i.e., the expected error probability of a policy, does not satisfy adaptive submodularity. Furthermore, it is nontrivial to understand how one can directly relate the two objectives: the ECED objective of Eq. (4.5.2), which we utilize for selecting informative tests, and the gain in the reduction of error probability, which we use for evaluating a policy.

We circumvent such problems by introducing surrogate functions, as a proxy to connect the ECED objective Δ_{ECED} with the expected reduction in error probability p_{ERR}. Ideally, we aim to find some auxiliary objective, denoted by f_{AUX}, such that the tests with the maximal Δ_{ECED} also have a high gain in f_{AUX}; meanwhile, f_{AUX} should also be comparable with the error probability p_{ERR}, such that minimizing f_{AUX} itself is sufficient for achieving low error probability.

We consider the function $f_{\text{AUX}} : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$, defined as

$$f_{\text{AUX}}(\psi) = \sum_{(\theta,\theta')\in\mathcal{E}} \mathbb{P}\left[\theta \mid \psi\right] \mathbb{P}\left[\theta' \mid \psi\right] \cdot \log \frac{1}{\mathbb{P}\left[\theta \mid \psi\right] \mathbb{P}\left[\theta' \mid \psi\right]} + c \sum_{y\in\mathcal{Y}} \mathbb{H}_2\left(\mathbb{P}\left[y \mid \psi\right]\right). \tag{4.5.3}$$

Here $\mathbb{H}_2(x) := -x\log x - (1-x)\log(1-x)$, and c is a constant that will be made concrete shortly (in Lemma 4.21). Interestingly, we show that function f_{AUX} is intrinsically linked to the error probability:

Lemma 4.20. *We consider the auxiliary function defined in Equation (4.5.3). Let $m \triangleq |\operatorname{supp}(\Theta)|$ be the number of root-causes, and $p_{\text{ERR}}^{MAP}(\psi)$ be the error probability given partial realization ψ. Then*

$$2c \cdot p_{\text{ERR}}^{MAP}(\psi) \leq f_{\text{AUX}}(\psi) \leq (3c+4) \cdot \left(\mathbb{H}_2\left(p_{\text{ERR}}^{MAP}(\psi)\right) + p_{\text{ERR}}^{MAP}(\psi)\log m\right).$$

We defer the poof of Lemma 4.20 to §A.2.6. The above result indicates that, if we can show that by running ECED, we can effectively reduce f_{AUX}, i.e., the one-step gain in f_{AUX} is significant in comparison with the gain of the optimal policy, then by Lemma 4.20, we can conclude that ECED also makes significant progress in reducing the error probability $p_{\text{ERR}}^{\text{MAP}}$.

Bounding the Gain w.r.t. the Auxiliary Function It remains to understand how ECED interacts with f_{AUX}. For any test v, we define

$$\Delta_{\text{AUX}}(v \mid \psi) \triangleq \mathbb{E}_{x_v}[f_{\text{AUX}}(\psi \cup \{v, x_v\}) - f_{\text{AUX}}(\psi) \mid \psi]$$

to be the expected gain of test v in f_{AUX}. Let $\Delta_{\text{EC}^2,\psi}(v)$ denote the gain of test v in the EC^2 objective, assuming that the edge weights are configured according to the *posterior distribution* $\mathbb{P}\left[\Theta \mid \psi\right]$. Similarly, let $\Delta_{\text{ECED},\psi}(v)$ denote the ECED gain, if the edge weights are configured according to $\mathbb{P}\left[\Theta \mid \psi\right]$. In §A.2.7 we prove the following result:

Lemma 4.21. *Let $m = |\operatorname{supp}(\Theta)|$, $n = |\mathcal{Y}|$, and ϵ be the noise rate associated with test $e \in \mathcal{V}$. Fix $\eta \in (0,1)$. We consider f_{AUX} as defined in Equation (4.5.3), with $c = 8\left(\log(2m^2/\eta)\right)^2$. It holds that*

$$\Delta_{\text{AUX}}(v \mid \psi) + c_{\eta,\epsilon} \geq \Delta_{\text{ECED},\psi}(v) \cdot (1-\epsilon)^2/16 = c_\epsilon \Delta_{\text{EC}^2,\psi}(v),$$

where $c_{\eta,\epsilon} = 2n(1-2\epsilon)^2\eta$, and $c_\epsilon \triangleq (1-2\epsilon)^2/16$.

Lemma 4.21 indicates that the test being selected by ECED can effectively reduce f_{AUX}.

Lifting the Adaptive Submodularity Framework Recall that our general strategy is to bound the one step gain in f_{AUX} against the gain of an optimal policy. To do so, we need to show that our surrogate exhibits, to some extent, the diminishing returns property. By Lemma 4.21 we can relate $\Delta_{\text{AUX}}(X_v \mid \psi_\pi)$, i.e., the gain in f_{AUX} under the *noisy* setting, to $\Delta_{\text{EC}^2,\psi}(X_v)$, i.e., the expected weight of edges *cut* by the EC^2 algorithm. Since f_{EC^2} is adaptive submodular, this allows us to lift the adaptive submodularity framework into the analysis. As a result, we can now relate the 1-step gain w.r.t. f_{AUX} of a test selected by ECED, to the cumulative gain w.r.t. f_{EC^2} of an optimal policy in the noise-free setting. Further, observe that the EC^2 objective at ψ satisfies:

$$f_{\text{EC}^2,\psi} := \sum_y \mathbb{P}\left[y \mid \psi\right]\left(1 - \mathbb{P}\left[y \mid \psi\right]\right) \overset{(a)}{\geq} 1 - \max_y \mathbb{P}\left[y \mid \psi\right] = p^{\text{MAP}}_{\text{ERR}}(\psi). \tag{4.5.4}$$

Hereby, step (a) is because the error probability of a MAP estimator always lower bounds that of a stochastic estimator (which is drawn randomly according to the posterior distribution of Y).

Suppose we want to compare ECED against an optimal policy OPT. By adaptive submodularity, we can relate the 1-step gain of ECED in $f_{\text{EC}^2,\psi}$ to the cumulative gain of OPT. Combining Equation (4.5.4) with Lemma 4.20 and Lemma 4.21, we can bound the 1-step gain in f_{AUX} of ECED against the k-step gain of OPT, and consequently bound the cost of ECED against OPT for Problem 4.5.1. We defer a more detailed proof outline to §A.2.5 and provide the full proof in detail in §A.2.9.

4.6 Experimental Results

In this section, we evaluate our algorithms on four sequential decision-making tasks, including

1. a Bayesian experimental design task intended to distinguish among economic theories of how people make risky decisions,
2. an active preference learning task via pairwise comparisons,
3. an adaptive troubleshooting task for mobile devices, and
4. an active touch-based localization task for robotic manipulation.

Table 4.2 summarizes how these problem instances fit into our framework.

Application	Test	root-cause	Decision
Pref. learning	movie pairs	favorable movie	e.g., genre
Behavior economics	lottery pairs	parameterized theory	valuation theory
Troubleshooting	symptoms	cause of problem	diagnosis
Touch-based localization	guarded move	target location	manipulation action

Table 4.2: Tests, root-causes, and decisions for different applications.

4.6.1 Overview of Experimental Setup and Baselines

Depending on the problem setup and the varying purposes of different tasks, we choose different algorithms to evaluate. In particular, the first two applications are simulations, and hence for these tasks, we can synthesize different experimental scenarios, and assess all the (submodular surrogate-based) objectives (including GBS, EC^2, HEC, DiRECt and ECED) discussed in this chapter. The troubleshooting application involves real-world diagnosis data collected from contact center agents, and we use it to showcase the performance of our dynamic hypotheses enumeration framework. Finally, the touch-based localization experiment is conducted on an actual robotic platform. We run both simulation and real-world experiments to demonstrate both the efficiency and effectiveness of the submodular surrogate-based objectives.

Baseline Approaches

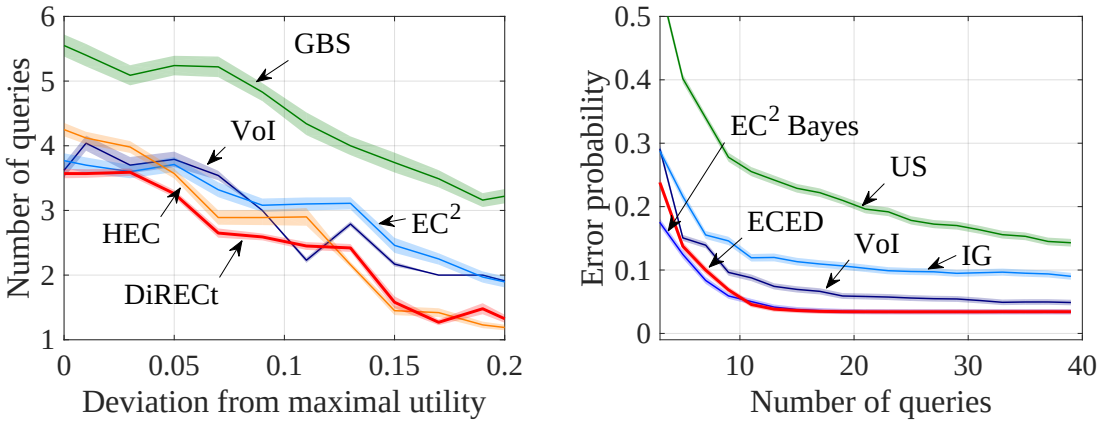
The DRD setting. In applications where the set of hypotheses are given (e.g., touch-based localization), we can directly run DiRECt without appealing to the sampling technique discussed in §4.4. We evaluate DiRECt against several baseline approaches for solving the DRD problem (Problem 4.1.3). The first baseline is the myopic decision-theoretic value of information algorithm (which chooses tests according to Eq. (4.2.2)), denoted as VoI. The second baseline is our recently proposed objective for addressing the DRD problem, HEC (§4.2.2). We also compare with GBS and EC^2. As discussed earlier, these two approaches are designed for solving special cases of the DRD problem. Assuming test outcomes are deterministic functions of hypotheses, the choices of tests by GBS and the most informative selection policy (which greedily picks the tests that are most informative about H) are in fact the same, assuming ties are broken in a deterministic way. This maximal informative selection policy is also known as

uncertainty sampling in active learning. When hypotheses are in multiple decision regions, EC^2 cannot be used as is. Hence, to run EC^2 we choose to randomly assign each hypothesis to one of the decision regions that it is contained in. For both GBS and EC^2 we use same stopping criteria as the DIRECT algorithm, i.e., we stop once one decision region contains all the consistent hypothesis.

The Noisy ECD setting. Under the noisy ECD setting, the hypotheses are not explicitly given, and we only have access to the prior distribution over the root-causes and test outcomes $\mathbb{P}[\Theta, X_1, \ldots, X_t]$. We consider running ECED, and evaluate it against the following baselines. The first one is EC^2-Bayes (as described in §4.5.2), which uses the Bayes' rule to update the edge weights when computing the gain of a test. Note that after observing the outcome of a test, both ECED and EC^2-Bayes update the posteriors on Θ and Y according to the Bayes' rule; the only difference is that they use different strategies when *selecting* a test. Also, we compare with two commonly used sequential information gathering policies: Information Gain (IG), and Uncertainty Sampling (US), which consider picking tests that greedily maximize the reduction of entropy over the target variable Y, and root-causes Θ respectively. Last, we also consider the myopic VoI policy (VoI), In our problems, VoI greedily picks the test maximizing the expected reduction in prediction error in Y.

4.6.2 Preference Elicitation in Behavioral Economics

The goal of our first application is to identify the underlying valuation theories people use when facing different choices of risky decisions, by sequentially asking test subject to choose between the various decisions. Several theories have been proposed in behavioral economics to explain how people make such decisions under and uncertainty. To set up the experiments, we consider six theories of subjective valuation of risky choices [Wak10; TK92; Sha64], namely, (1) *expected utility with constant relative risk aversion*, (2) *expected value*, (3) *prospect theory*, (4) *cumulative prospect theory*, (5) *weighted moments*, and (6) *weighted standardized moments*. Here, each theory corresponds to a possible value of the target variable Y in the DRD problem. To construct the test set $\mathcal{V}$, we consider *lotteries*: a lottery L is a known distribution over payoffs (e.g., the monetary value gained or lost). Each test consists of a pair of lotteries, and the outcome of the test is the choice made between the two by the test subject. root-cause



(a) *DRD* (b) *Noisy ECD*

Figure 4.14: Results for the experimental design task in behavioral economics.

Θ corresponds to a parameterized theory that predicts, for a given test, which lottery is preferable. The goal, is to adaptively select a sequence of tests to present to a human test subject to distinguish which of the six theories best explains the subject's responses (i.e., predicting the target variable Y).

Evaluating DiRECt. To generate decisions, root-causes and tests, we employ a similar procedure as suggested in the Bayesian experimental design literature [Ray+12]: First, we assume a uniform distribution over the 6 type of theories (i.e., decisions) Y. Then, we instantiate each theory using a grid of parameters, and use these parameterized theories as root-causes Θ. Parametrized theories of the same type are assigned with the same prior probabilities and therefore we obtain a set of parameterized theories with a non-uniform prior distribution.

We generated $\sim$ 16K pairs of lotteries as tests. Given root-cause θ and test $v = (L_1, L_2)$, one can compute the values of lottery L_1 and lottery L_2, denoted by u_1 and u_2. Assuming test outcomes are noise-free, X_v is thus defined as $X_v := \mathbb{1}\{u_1 \geq u_2\}$. Clearly, given a parameterized valuation theory θ, the outcome of a test is deterministic: $\mathbb{P}[X_v \mid \theta] \in \{0, 1\}$. In other words, in the *noise-free* setting, each root-cause θ corresponds to fixed realization of all tests $\mathbf{x}_{\mathcal{V}}$, and hence corresponds to a hypothesis in the DRD problem.

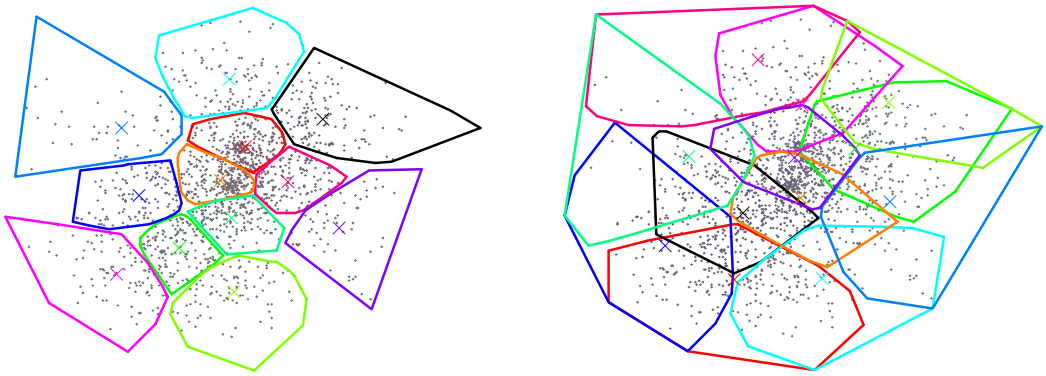
In our experiments, we allow a tolerance ϵ - that is, if one hypothesis differs from another by at most ε, they are considered to be similar, and thus have the same set of

optimal decisions. Here, by "difference" we mean the normalized edit distance between two hypotheses: if there are t tests, then $d(h_1, h_2) = \frac{|\{i: f_i(h_1) \neq f_i(h_2)\}|}{t}$. Hypotheses that are at most ε apart are considered to share a decision region. Results for simulated test outcomes with varying ε are shown in Figure 4.14a. As we can observe, when varying ε, the *query complexity* (i.e., the number of queries needed to identify the target region) of DIRECT is lower than all baselines in most of the scenarios. Due to the lack of consideration of (overlapping) decision regions, both GBS and EC^2 perform worse than HEC and DIRECT. The performance of the myopic VoI policy is rather unstable. As our theory suggests, such posterior based myopic selection policy can perform quite badly.

Evaluating ECED. We further study the Noisy ECD setting, where the feedback on test outcomes are considered to be *noisy*. To model noisy test outcomes, we consider the Bradley-Terry-Luce (BTL) preference model [BT52], where the probability that root-cause θ favors lottery L_1 over lottery $L_2$is defined as $\mathbb{P}[X_e = 1 \mid \theta] = \frac{1}{1+\exp(-\lambda \cdot (u_1 - u_2))}$, where λ controls the level of noise in the system. The BTL model has been widely for pairwise data, e.g., [NOS12; Sha+15], etc. Intuitively, a user is more prone to error if the utilities of a pair are close. For preference elicitation, imagine a pair of lotteries (L_1, L_2) which is almost of equal value to the user, then her feedback on whether she favors L_1 over L_2 is very noisy.

In the Noisy ECD setting, we do not need to set the tolerance parameter ε. Rather, we assign each parameterized theory to the decision region representing the corresponding theory type and assume that we can tolerate some edges not being cut in the end (because our goal is to identify the target theory with low error probability).

To compare different algorithms, we set a budget on the number of iterations allowed, and plot the error probability as a function of the number of iterations. Fig. 4.14b demonstrates the performance of ECED, with $\lambda = 10$. The average error probability has been computed across 1000 random trials for all methods. We observe that ECED and EC^2-Bayes have similar behavior on this data set; however, the performance of the uncertainty sampling algorithm (US) is much worse. This can be explained by the nature of the data set: due to the way we assign the prior probability, the data set has a more concentrated distribution over Θ, but not Y. Therefore, since tests only provide indirect information about Y through Θ, what the uncertainty sampling scheme tries to optimize is Θ. Hence it performs quite poorly.



(a) *Partitions* ($r = 12$) (b) *Assign each movie to 2 cluster centroids*

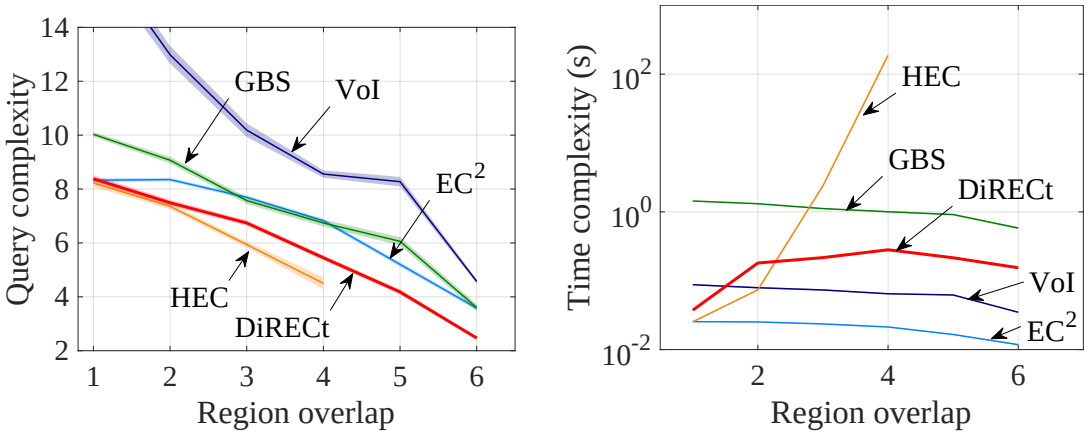
Figure 4.15: A 2-d illustration of (overlapping) decision regions for *MovieLens* experiments. Dots represent movies; cross markers represent cluster centroids, and colored polygons represent decision region boundaries. (a) Movies are partitioned into 12 disjoint clusters. (b) Each movie is assigned to the two closest centroids.

4.6.3 Preference Learning via Pairwise Comparisons

Our second application concerns a comparison-based movie recommendation system, which learns a user's movie preference (e.g., the favorable genre) by sequentially showing her pairs of candidate movies, and letting her choose which one she prefers.

Constructing decision regions We use the *MovieLens 100k* dataset [Her+99], which consists a matrix of 1 to 5 ratings of 1682 movies from 943 users. To measure the similarity of candidate movies, we extract movie features by computing a low-rank approximation of the user/rating matrix of the *MoiveLens 100k* dataset through singular value decomposition (SVD). Specifically, we extract a 10-dimensional feature vector for each movie. We then use k-means to partition the set of movies into r (non-overlapping) clusters in the Euclidean space, which simulate the decision regions. We choose the k-means cluster centers as the centroids representing the "categories" that a user may be interested in. Since one movie can usually belong to several categories, we assign each movie to the category that is represented by the closest centroids, giving us overlapped decision regions. See Fig. 4.15 for an illustration.

Generating tests. Tests are given in the form of movie pairs. There is a total number of $\approx$1.4 million pairs of movies in the *MoiveLens 100k* dataset. Usually, to distinguish a



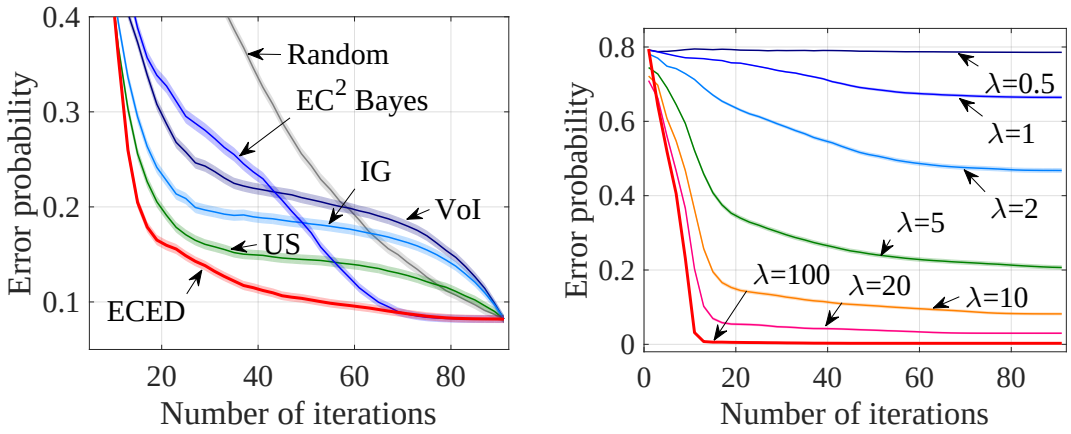
(a) *DRD: MovieLens - Label Complexity* (b) *DRD: MovieLens - Time Complexity*

Figure 4.16: Results for the active preference learning application.

movie from the rest of the pool, we don't need to perform all the tests. Rather, we want to extract a subset of tests from all the available tests, such that by performing this subset of tests, one can uniquely distinguish all the movies in our pool. To select this subset, we first build a binary matrix $A = \{a_{i,j}\}_{1682\times1682}$ of size 1682×1682, representing all pairs of movies to be distinguished. If performing a test v can distinguish a pair of target movies indexed by(i, j) (meaning that by performing the test, one can tell which one of (i, j) is more favorable), then we fill the entry $a_{i,j} = 1$, indicating that we can distinguish i from j by performing this test. Therefore, we start from an empty set T of tests and keep adding tests (following some random order) into the T till the matrix A is filled up. This amounts to a total number ≈ 100 tests, with which we can uniquely identify any of the 1682 movies.

Evaluating DiRECt. Under the DRD setting, each movie corresponds to a hypothesis. After receiving feedback from a test, we remove the movies that we believe do not reflect user's interest (e.g., movies that are more similar to the one that the user chooses to dislike). Once all the remaining movies in our pool are in the same category (i.e., *decision region*), we can recommend any of the movies to the user. Our goal is to identify such a category by asking as few pair-wise comparison questions as possible.

We demonstrate the performance of DiRECt on *MovieLens* in Figure 4.17. We fix the number of clusters (i.e., decision regions) to 12, and vary r, the number of assigned regions for each hypothesis, from 1 to 6. Note that r controls the hyperedge cardinality



(a) *Noisy ECD: MovieLens* (b) *Noisy ECD: Varying noise*

Figure 4.17: Results for ECED on *MovienLens*.

in HEC, which crucially affects the computational complexity. As we can observe, the *query complexity* (i.e., the number of queries needed to identify the target region) of DIRECT is lower than all baselines except HEC. However, it is significantly faster to compute. See Figure 4.16b (for $r = 5$, HEC failed to pick any tests within an hour, and eventually ran into memory issue when we continued running the experiment).

Evaluating ECED. Under the Noisy ECD setting, we make an (perhaps overly simplified) assumption that each movie only belongs to one category (i.e., $r = 1$). As with the previous application, we assume that test outcomes are noisy, and we model the noise with the BTL model: for a given test $v = (m_a, m_b)$ and root-cause (i.e., favorite movie) θ, we assume $\mathbb{P}[X_v = 1 \mid \theta] = \frac{1}{1+\exp(-\lambda \cdot (d(m_a,\theta) - d(m_b,\theta)))}$. Hereby, $d(\cdot,\cdot)$ is the distance function, and λ controls the noise level: larger λ corresponds to a steeper sigmoid function (and hence low noise).

Fig. 4.17a shows the performance of ECED compared other baseline methods when we fix the size of $\mathcal{Y}$ to be 20 and λ to be 10. We compute the average error probability across 1000 random trials for all methods. We can see that ECED consistently outperforms all other baselines. Interestingly, EC^2-Bayes performs poorly on this data set. This may be because the noise level is still high, misguiding the two heuristics to select noisy, uninformative tests. Fig. 4.17b shows the performance of ECED as we vary λ. When $\lambda = 100$, the tests become close to deterministic given a root-cause, and ECED can achieve 0 error with ~ 12 tests. As we increase the noise rate (i.e., decrease λ), it

takes ECED many more queries for the prediction error to converge. This is because, with high noise rate, ECED discounts the root-causes more uniformly. Hence they are hardly informative in Y. This comes at the cost of performing more tests, and hence low convergence rate.

4.6.4 Interactive Troubleshooting

We further examine the submodular-surrogate based framework on a real-world interactive troubleshooting platform. Our data is collected from contact center agents and knowledge workers who solve complex troubleshooting problems for mobile devices. These training data involve around 1100 root-causes (the possible values of the root-cause Θ) and 950 tests (questions on symptoms customers may encounter) with binary outcomes. From the training data, we derived a joint distribution over $[X_1, \ldots, X_t]$ and Θ as $\mathbb{P}[x_1, \ldots, x_t, \theta] = \mathbb{P}[\theta] \prod_{i=1}^{n} \mathbb{P}[x_i \mid \theta]$, where $\mathbb{P}[\theta]$ is the prior distribution over the root-causes, which we assume to be uniform.

We simulated over 10,000 test scenarios (10 scenarios for each θ), where a customer enters the system with an initial symptom x_{v_0} (i.e. a test outcome), according to probability $\mathbb{P}[x_{v_0} \mid \theta]$. Each scenario corresponds a to a root-cause θ and an underlying hypothesis h. The number of decisions is the number of root-causes (which correspond to making a diagnosis), plus one extra decision of "give-up". Intuitively, if two root-causes result in the same outcomes for all tests, then the virtual agent cannot decide which one is the true root-cause and therefore will forward such case to a human agent, corresponding to the "give-up" decision. In practice, introducing such "give-up" decision guarantees that there are no overlaps between decision regions. The utility function $u(\theta, y)$ corresponds to the cost of misprediction (i.e., the cost of mispredicting a root-cause, or the cost of "give-up"), which is specified by the business *domain expert* as

$$u(\theta, y) = \begin{cases} 0, & \text{if } y \text{ is "give-up";} \\ 1, & \text{if } y \text{ matches } \theta, \text{ i.e., } y \text{ is the correct decision} \\ -19, & \text{if } y \in \text{supp}(\Theta) \wedge y \neq \theta, \text{ i.e., } y \text{ corresponds to a wrong root-cause} \end{cases}$$

In this way, the "give-up" decision is optimal when the posterior distribution over Y given *all* test outcomes have no "peak" value higher than 95%.

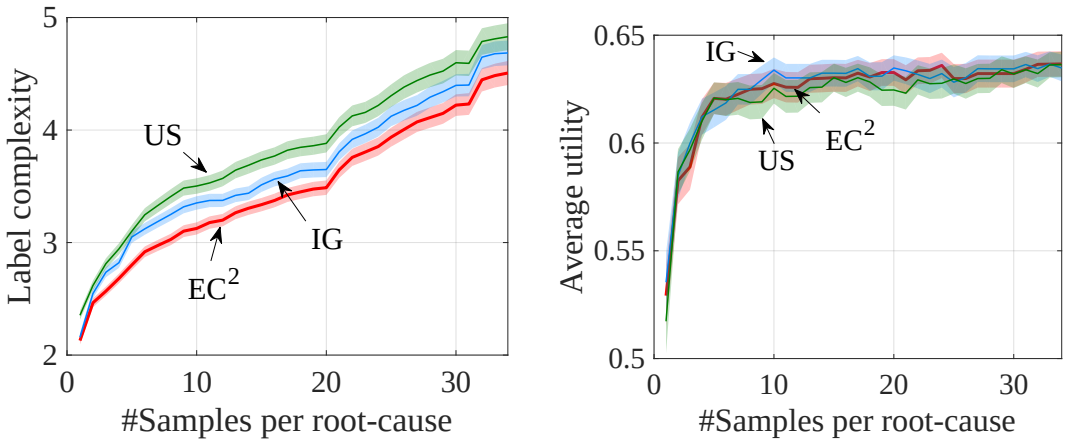


Figure 4.18: Results: Interactive troubleshooting.

We compared four algorithms: EC^2, myopic Value of Information (VoI), Information Gain (IG), and Uncertainty Sampling (US). Fig. 4.19 shows the expected entropy on Θ while increasing the test budget. Clearly, myopic VoI performs comparably worse than others. In Fig. 4.18, we report the average queries complexity (i.e., the expected number of tests required to solve a case), and the average utility of making decisions while varying the maximal number of samples allowed for each root-cause. We see that as we sample hypotheses more extensively, all algorithms require more tests to make a decision; on the other hand, the quality of decisions also increases with more samples. This behavior is reasonable since having too few samples excludes a lot of good candidates which in turn leads to poor utility. Moreover, there is a $\sim$16% improvement in the average query complexity when using the EC^2 algorithm. This shows a clear advantage of using submodular surrogates for this kind of sequential problem: EC^2 by construction is "less myopic".

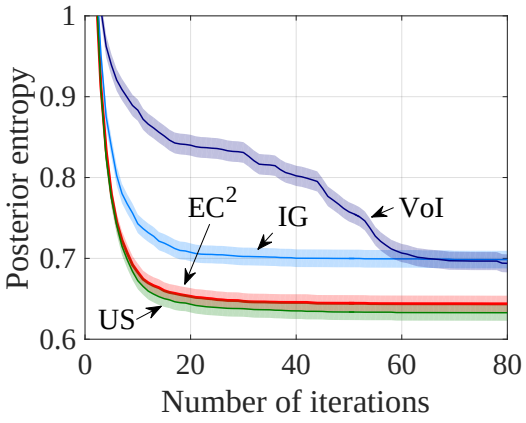
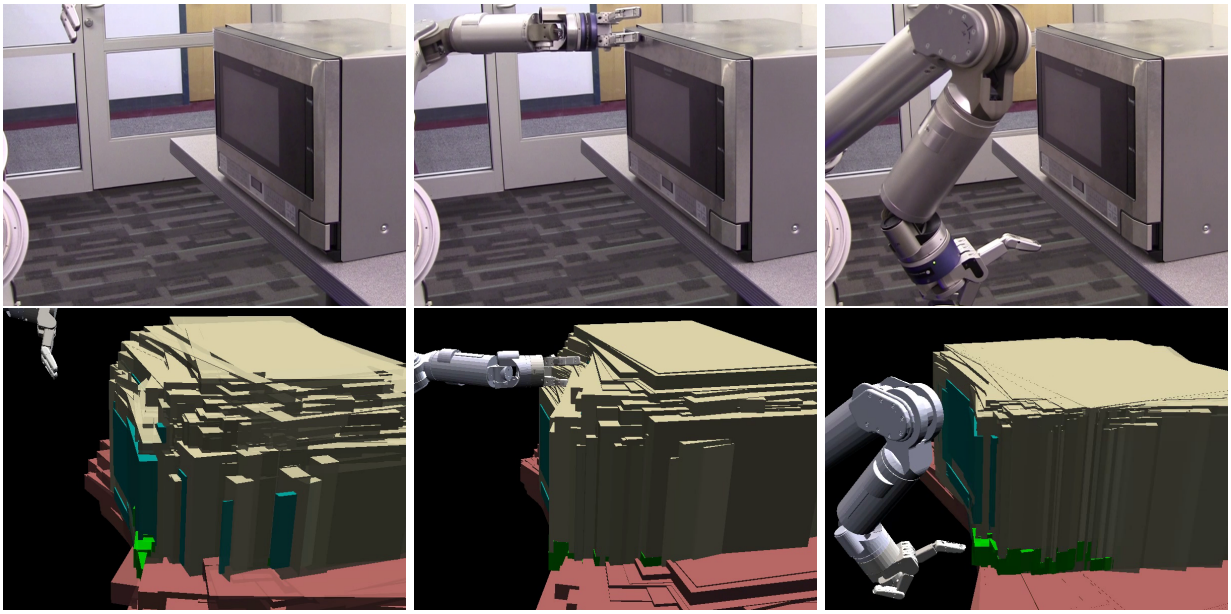


Figure 4.19: Posterior entropy.

4.6.5 Active Touch-based localization

Our fourth application is a robotic manipulation task of pushing a button, with uncertainty over the target's pose (See Fig. 4.20). Tests consist of *guarded moves* [WG75], where the end effector moves along a path until contact is sensed. Those hypotheses

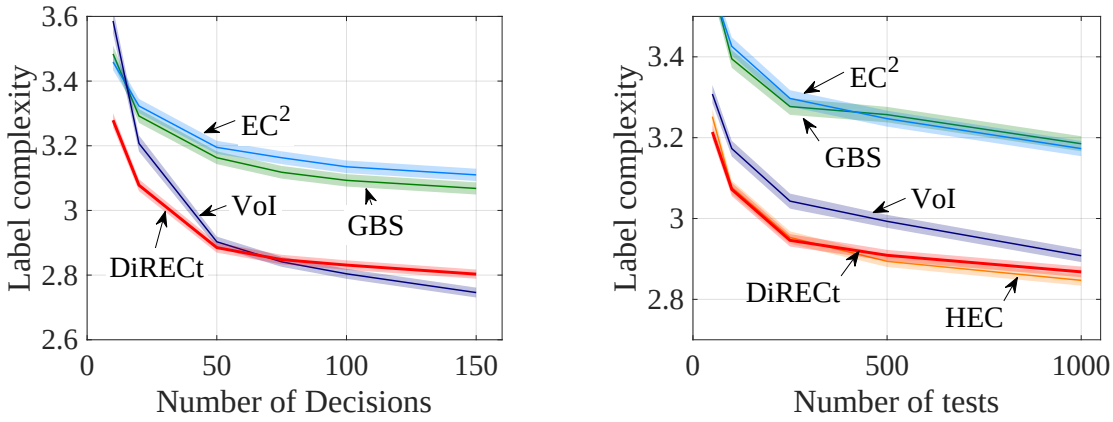
which would not have produced contact at that location (e.g., they are far away) can be eliminated. Decisions correspond to putting the end effector at a particular location and moving forward. The coinciding decision region consists of all object poses where the button would successfully be pushed. Our goal is to concentrate all consistent hypotheses within a single decision region using the fewest tests.



(a) Hypotheses (b) Tests (c) Decision regions

Figure 4.20: Experimental setup for touch-based localization. (a) Uncertainty is represented by hypotheses over object pose. (b) Tests are guarded moves, where the end effector moves along a path until contact is sensed. Hypotheses which could not have produced contact at that location (e.g. they are too far or too close) are removed. (c) Decisions are button-push attempts: trajectories starting at a particular location, and moving forward. The corresponding region consists of all poses for which that button push would succeed.

Generating root-causes / hypotheses. To model the pose uncertainty of the target object, we use four parameters: (x, y, z) for positional uncertainty, and ν for rotation about the z axis. An initial set of 20000 hypotheses are sampled from a normal



(a) *Robot* - varying $|\mathcal{Y}|$

(b) *Robot* - varying $|\mathcal{V}|$

Figure 4.21: Results: Touch-based localization.

distribution $N(\mu, \Sigma)$, where μ is some initial location (e.g., from a camera), and Σ is diagonal with $\sigma_x = \sigma_y = \sigma_z = 2.5$cm, and $\sigma_\nu = 7.5°$.

We run DIRECT on both simulated data and a real robot platform. When running on a real robot, many actions are infeasible due to kinematic constraints. In the first simulated experiment, we fix the set of guarded moves $|\mathcal{V}| = 250$, and evaluate the algorithms by varying the number of randomly sampled decisions $|\mathcal{Y}|$. Sampling decisions enables us to generate arbitrarily many, ensuring we always have many decisions available. To compute the *myopic* (VoI), we define a utility function $u(h, \mathcal{R})$ which is 1 if $h \in \mathcal{R}$ and 0 otherwise. Results are plotted in Fig. 4.21a. We see that DIRECT generally outperforms other baselines. Here, myopic VoI performs comparably – likely because the problem is solved within a short horizon.

For the results shown in Fig. 4.21a, it's prohibitive to run HEC, because the overlap between regions is large. We also want to compare DIRECT with HEC on problem instances where HEC can practically run. To ensure that, we pre-select a grid of 25 button pushing actions $\mathcal{Y}$ while ensuring the overlap is minimal, so that the HEC objective can be computed in a reasonable time. Note that to run DIRECT, we don't need to enforce such strict constraints. We randomly generate guarded moves $\mathcal{V}$ to select from. In Fig. 4.21b we show the number of test guarded moves needed for different algorithms, when varying $|\mathcal{V}|$. As we can observe from the results, DIRECT performs essentially the same as HEC on this problem instance, while slightly outperforms VoI.

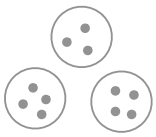
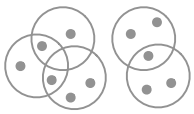
Setting	Submodular surrogates	"Noisy" extension
ODT 	GBS $O\left(\log \frac{1}{p_{\min}}\right)$	**MIS** (§3.3.1) $O\left(\frac{\log n}{(1-2\epsilon^2)}\right)$
ECD 	EC^2 $O\left(\log \frac{1}{p_{\min}}\right)$	**ECED** (§4.5) $O\left(\frac{(\log n/\delta)^3}{(1-2\epsilon^2)}\right)$
DRD 	**HEC / DiRECt** (§4.3) $O\left(\log \frac{1}{p_{\min}}\right)$	–

Table 4.3: Summary of results for NVOI-NMU. Our results are highlighted in **bold**. For the results displayed under the "noisy" extension, we assume binary test outcomes with independent flips of the label and compare with the "optimal" policies under some stronger stopping condition.

4.7 Summary

In this chapter, we studied practical and efficient approaches for optimizing value of information. We described a class of algorithms which rely on adaptive submodular surrogates and proved strong theoretical guarantees. We presented DiRECt and proved its objective function to be adaptive submodular, making it amenable for efficient greedy optimization. To accommodate the computational issue that DiRECt could run into in many real-world problems, we proposed a novel hypothesis sampling strategy on top of the DiRECt framework and showed that it compensated the inefficiency of the popular submodular surrogates-based greedy algorithms, while still enjoying provable theoretical guarantees. We demonstrated the effectiveness of our (submodular surrogate-based) algorithmic framework extensively on several applications. In our applications, DiRECt compares favorably with existing approaches, while being significantly faster than competing methods.

Despite the technical merit of the submodular surrogate-based methods, the bounds for these algorithms can not handle general noise on the test outcomes. To this end, we introduced ECED, which strictly generalizes the EC^2 algorithm to the noisy

setting (which we call the noisy ECD problem). We proved that ECED enjoyed strong theoretical guarantees when comparing with an optimal policy achieving some lower error probability. We also demonstrated the compelling performance of ECED on two (noisy) problem instances.

Table 4.3 summarizes our theoretical results presented thus far. The analysis framework presented in this chapter does not apply to the general problem which involves both overlapping decision regions and noisy observations. However, in practice, one can still run edge discounting algorithm similarly with ECED. We believe that our work makes a major step towards building towards building practical and robust sequential decision-making systems, and provides useful insight for understanding the theoretical aspects of adaptive information acquisition algorithms in complex, noisy environments.

5

Active Object Detection

In §4, we have demonstrated that it is useful to use adaptive submodular surrogates for the broad class of Bayesian active learning and experimental design problems. Our key idea was to reduce the non-submodular optimization problems (e.g., decision-theoretic value of information) to submodular ones (e.g., DRD) by reformulating the problems. This is somewhat similar to the idea of constructing convex surrogates in the statistical learning literature. However, unlike their continuous analog, submodular surrogates have been much less attended to in the literature. An immediate follow-up question from the previous chapter is, can we generalize this idea, and construct adaptive submodular surrogates for other sequential decision problems?

In this chapter, we present a novel application that goes beyond Bayesian active learning, where adaptive submodular surrogate leads to promising performance. In particular, we consider the *object detection* problem – a fundamental challenge in computer vision. While a human expert is quite capable of recognizing visual objects in a scene, for many practical tasks, a high-quality fully automatic detection system is still beyond our reach. A major problem for automatic object detection is the lack of sufficient training examples since manual annotations are usually time-consuming and expensive. To illustrate this, let us consider the task of detecting orangutan nests in aerial photos, as described below.

Example 5.1. *Ecologists launch micro UAVs, "conservation drones" to take high-quality photographs of orangutan habitat from above treetops, to obtain accurate and timely data on*

Figure 5.1: Orangutan nests detection for biodiversity monitoring (UAV-FOREST). (Left) Conservation drone (image courtesy of conservationdrones.org). (Middle) An aerial image captured by the conservation drone, with two orangutan nests highlighted. (Right) The response image (in grayscale) generated by a base detector.

orangutan distribution in the surveyed area. On the one hand, frequently going through thousands of those photographs to look for orangutan nests, is an extremely tedious and potentially expensive task for human experts. On the other hand, an automatic detection system (e.g., the rightmost figure in Figure 5.1) tends to produce many false positive detections in the high-clutter background, given limited training samples obtained from the drone missions.

A natural step towards a sustainable and efficient object detection system is to introduce human supervision along with the automatic detection process. In such settings, the system and the user collaborate to obtain the best performance: first, the system proposes candidate objects to the expert for verification, and then the expert provides feedback to guide the system to generate better detections. To make the best use of the scarce labeling resources, one needs to decide in which order to query the candidates, such that the best possible performance could be achieved in exchange for the minimum amount of user supervision.

Our contribution. Rather than developing novel object recognition algorithms, in this chapter, we focus on techniques for intelligently interacting with humans. We assume that we already have access to a base detector/ classifier that can produce a certain response for target objects (c.f. Fig. 5.1), and the task is to apply the classifier to the multiple object detection problem while having humans in the loop. We propose a general framework for such *active detection* problems, which brings together the quality of manual annotation and the scalability and speed of automatic detection, regardless of what base detectors have been employed. We show how one can, from a given base detector, derive a natural sequential decision problem. Further, its objective function

is adaptive submodular, allowing us to employ the adaptive greedy optimization framework, with strong theoretical guarantees. To demonstrate the effectiveness of active detection, we carry out experiments on three different detection tasks using different base object detectors and show that active detection does have substantial advantages over its passive counterpart. To the best of our knowledge, our approach is the first to rigorously address the active detection problem from both empirical and theoretical perspectives.

Organization of this chapter. We formulate active detection as a sequential decision making problem in §5.1. In §5.2, we present ActDet, a principled greedy detection framework, and state the main theoretical results. In §5.3, we instantiate ActDet for three different real-world active detection tasks and demonstrate its superior performance over the passive approaches and some active baselines. We review the background on object detection in §5.4 and summarize this chapter in §5.5.

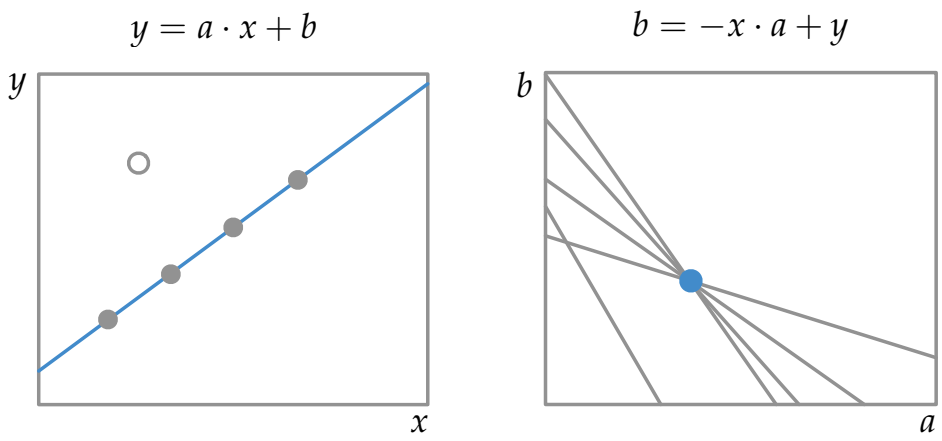
5.1 Problem Statement

We start by introducing a simple Hough-based detection algorithm that motivates our research. Then, we formalize the active detection problem in section 5.1.2.

5.1.1 Hough-based Approaches

Hough-based detection algorithms work by transforming the input image into a new representation in a domain called the Hough space [Hou59; Bal81]. Each point in the Hough space corresponds to a possible configuration of an object instance. We call such configuration a *candidate hypothesis*, or "candidate" in short, to avoid confusion with the usage of term "hypothesis" in §4.

The Hough image is built by aggregating the contributions of the individual *voting elements*, taken from the image or some appropriate set of features of it. The detections will then be identified as peaks in the Hough image, with the height of the peak as an indicator of the confidence in the detection. As an example, to detect lines in an image, one can search through the peaks in the 2-d Hough space (with each axis corresponding to one parameter of the line function), and find a subset of line parameters that have



(a) Original image (b) Hough image

Figure 5.2: Illustration of a hough-based line detector.

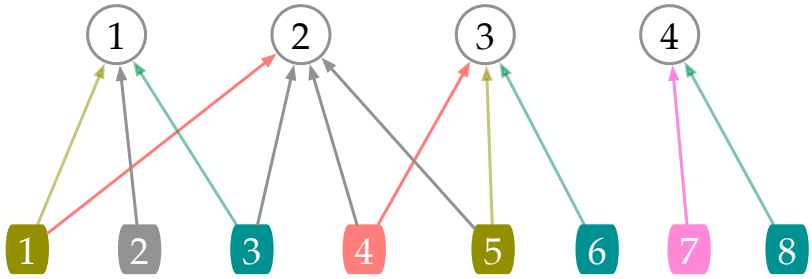
the highest accumulated votes (See Fig. 5.2 for an illustration). Similarly, to detect natural objects, one needs to create individual voting elements that vote for a certain configuration of the whole object.

5.1.2 Active Detection as a Sequential Decision Problem

We can generalize the idea of *voting elements* and *candidates* for general object detection problems. To align our notations with previous chapters, we use $\mathcal{V} = \{1, \ldots, t\}$ to denote the set of candidates, where each candidate $v \in \mathcal{V}$ represents a possible configuration (e.g., location, aspect ratio, size, etc.) of the target object. Later on, we will see that the candidate set $\mathcal{V}$ corresponds to the set of tests in the adaptive greedy optimization framework discussed in §2.3.

We use $X_1, \ldots, X_t \in \mathcal{X} = \{+1, -1\}$ to denote the (initially unknown) labels of the candidates, such that $X_v = +1$ if candidates v is true (i.e., there exists an object at location $\mathbf{z}_v$), and $X_v = -1$ otherwise. We use $\mathbf{X}_\mathcal{V} = \{X_1, \ldots, X_t\}$ to refer to the collection of all variables. Whenever a candidate v is selected, the corresponding variable x_v is revealed. Similarly, if we select a set of candidates $\mathcal{A}$, the corresponding observations are represented as $\mathbf{x}_\mathcal{A} \subset \mathcal{V} \times \mathcal{X}$.

We further assume a finite set of voting elements, denoted as Σ. We refer to Σ as the *evidence set*. Each item $\sigma \in \Sigma$ corresponds to a voting element that can cast votes for a set of candidates. A base object detector proposes a voting scheme that connects the



(a) *Binary votes*

Figure 5.3: The voting scheme proposed by a base object detector as a bipartite graph. "Similar" edges share the same color.

the evidence set Σ and the candidate set $\mathcal{V}$. The interaction between voting elements and candidates can be formally represented as a bipartite graph $\mathcal{G}(\Sigma, \mathcal{V}, \mathcal{E})$; each edge $(\sigma, v) \in \mathcal{E}$ is assigned a score (e.g., confidence, probability estimation given by the base object detector) with which σ votes for candidate v. We will give concrete examples in §5.3.

Active Detection as a Sequential Decision Problem We consider a sequential strategy, where the detector proposes a candidate $v \in \mathcal{V}$, and receives a label x_v from the expert. Whenever a label is revealed, we update the underlying bipartite graph, which represents the current state of the base detector. In particular, we perform the updates by only reducing the weights associated with the voting elements (i.e., covering the edges in $\mathcal{G}$), as observations will keep explaining the votes proposed by the base detector. Our goal, therefore, is to propose a strategy that can cover the entire set of edges as soon as possible.

5.2 The Active Detection Framework

We begin with the case where the votes generated by the base detector only consist of "yes" or "no" answers and then generalize to the setting with real-valued votes. In §5.2.3, we provide an efficient greedy solution to the active detection problem and present our main results.

5.2.1 Binary Votes Setting

We first study a simple case, where the voting elements can only cast binary votes (i.e., 0/1) for the presence of an object with candidate configuration/location encoded by $v \in \mathcal{V}$.

Suppose the active learner proposes a candidate, and receives a positive label from an external expert. Denote this candidate by $candidate_+$. Since a voting element has equal confidence in all its supporting candidates, the true candidate then thoroughly explains the voting elements $\sigma \in \Sigma$ that voted for v, thereby "covering" all votes associated with σ. We refer to the amount of edges covered by selecting v_+ with positive feedback as the *positive coverage* of v_+.

A more interesting case is when the active detector makes a false prediction. Let the *negative coverage* be the reduction of edge weights in $\mathcal{G}$ incurred by a false detection, denoted by $v_- \in \mathcal{V}$. The construction of negative coverage is akin to that of positive coverage, but with substantial difference: while in the positive case we cover the edges which are *neighboring* the edges that directly vote for v_+ (i.e., stemming from the same voting elements), in the negative case we will reduce the weight of all edges that are *similar* to the ones pointing to the false candidate v_-. Concretely, we assume that the votes generated by the base detector are associated with some features, and thereby can be clustered accordingly. The clustering associated with the base detector is denoted using a function $\texttt{cls} : \Sigma \times \mathcal{V} \to \mathbb{Z}^+$, which maps an edge $(\sigma, v) \in \mathcal{E}$ to its cluster index. A false detection thereby "explains away" (covers) similar votes that share the same clusters with the potentially false vote(s). See Figure 5.4 for an illustration.

Formally, the fraction of an edge $(\sigma, v) \in \mathcal{E}$ being covered due to negative observations, could be modeled as a monotone increasing function $g : \mathcal{E} \times 2^{\mathcal{V}\times\mathcal{X}} \to [0,1]$. In particular, we express it as a concave function q of the number of negative supports observed:

$$g(\sigma, v, \mathbf{x}_\mathcal{A}) = q(n_{\texttt{neg}}(\sigma, v, \mathbf{x}_\mathcal{A}))$$

where $n_{\texttt{neg}}(\sigma, v, \mathbf{x}_\mathcal{A}) \equiv |\{(v', -1) \in \mathbf{x}_\mathcal{A} : \exists \sigma', \texttt{cls}(\sigma', v') = \texttt{cls}(\sigma, v)\}|$ is the number of false candidates that are being voted for by any edge in the same cluster as (σ, v). In general, we want $q(0) = 0$ and $\lim_{n'\to\infty} q(n') = 1$. An extreme choice would be $q(n') = \min(n', 1)$: the edge is fully covered as soon as it is in the same cluster of a vote for a negative candidate. A less aggressive choice of the concave function, which

we adopt in our experiments, is:

$$q(n') = 1 - \gamma^{n'}. \tag{5.2.1}$$

The negative discount factor γ controls the speed with which the weights will be discounted. If $\gamma = 0$, all the edges in the cluster c will be fully discounted once one of them votes for a negative candidate; if $\gamma = 1$, the edges will never be discounted.

Now we are ready to construct the *coverage function* $f^{(1)}_{\sigma,v} : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$ for any edge $(\sigma, v) \in \mathcal{E}$, in the binary votes setting. Given a set of candidates $\mathcal{A} \subseteq \mathcal{V}$, and corresponding observations $\mathbf{x}_\mathcal{A} \subseteq \mathcal{V} \times \mathcal{X}$, the amount by which a given edge (σ, v) is covered is defined as

$$f^{(1)}_{\sigma,v}(\mathbf{x}_\mathcal{A}) = \max \begin{cases} 1, & \text{if } \exists v' : (v', +) \in \mathbf{x}_\mathcal{A} \wedge (\sigma, v') \in \mathcal{E}; \\ g(\sigma, v, \mathbf{x}_\mathcal{A}), & \text{otherwise.} \end{cases} \tag{5.2.2}$$

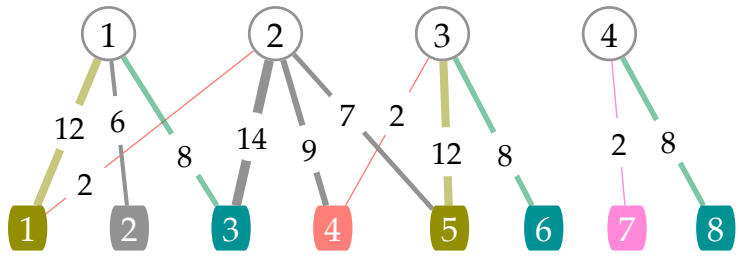
5.2.2 The General Case with Real-value Votes

The previous approach is limited, in a way that it only allows us to describe the support given by a voting element to a candidate as a binary relation. In realistic settings, we would like to take the strength of confidence into account, i.e., each edge (σ, v) is associated with a weight $w_{\sigma,v} \in \mathbb{R}_{\geq 0}$. For this more general scenario, we need to redefine "coverage", by allowing edges to be partially covered. Following the previous example, when an edge (σ, v) is covered due to positive observation, it will also cover its neighbors in a magnitude that is at most its weight $w_{\sigma,v}$. Since we do not allow negative weights, if a neighbor edge (σ, v') has a weight $w_{\sigma,v'} < w_{\sigma,v}$, then it is fully covered. Thus, an edge (σ, v) covers another edge (σ, v') in a magnitude given by $\min(w_{\sigma,v}, w_{\sigma,v'})$.

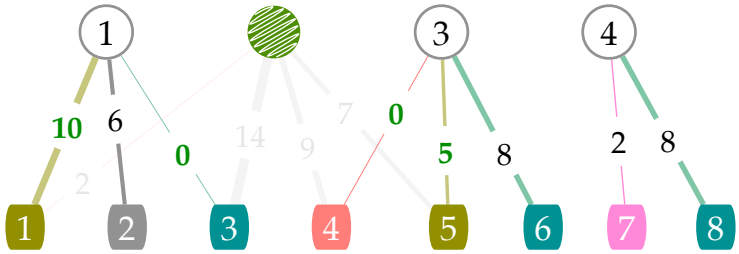
Taking negative coverage into account, the coverage function for an edge is defined as:

$$\begin{aligned} f_{\sigma,v}(\mathbf{x}_\mathcal{A}) &= g(\sigma, v, \mathbf{x}_\mathcal{A}) \cdot w_{\sigma,v} \\ &+ \min \left\{ \max_{v':(v',+1)\in\mathbf{x}_\mathcal{A}} w_{\sigma,v'},\ (1 - g(\sigma, v, \mathbf{x}_\mathcal{A})) \cdot w_{\sigma,v} \right\} \end{aligned} \tag{5.2.3}$$

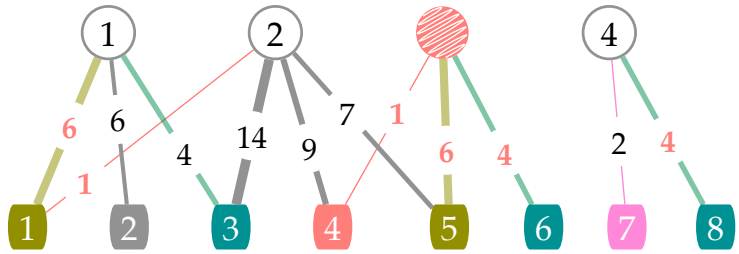
We can interpret the first term on the RHS as the fraction of weight covered due to negative observations, and the second term as the fraction of remaining weight (i.e., after negative discount) covered due to positive observations. Note that $w_{\sigma,v'}$ does not have a discount factor, since we know that the edge (σ, v') represents the vote for a positive candidate, and thus it should be fully covered.



(a) *Votes with real values*



(b) *(Positive) edge coverage*



(c) *(Negative) edge coverage*

Figure 5.4: (a) shows a bipartite graph with real-valued edge weight. (b) illustrates how we should update the graph from a positive feedback: if candidate 2 is true, then all the direct votes and their neighboring votes are (partially) covered; (c) illustrates how we should update the graph from a negative feedback: if candidate 3 is false, then all the "similar" edges as highlighted are (partially) covered.

Connection with the binary votes setting. We can see that the coverage function with binary votes (Eq. (5.2.2)) is a special case of the general coverage function (Eq. (5.2.3)), when all non-zero weights are set to 1: Assume the edge (σ, v) exists, i.e., $w_{\sigma,v} = 1$. If the maximum among the weights $w_{\sigma,v'}$ of the first term is zero, then the first term vanishes and we are left with $g(\sigma, v, \mathbf{x}_{\mathcal{A}})$. Note that all the $w_{\sigma,v'}$ being zero is equivalent to the second case of Eq. (5.2.2). The only alternative is if the maximum of the $w_{\sigma,v}$ is 1. Then, since $1 - g \leq 1$, we have that $f_{\sigma,v}(\mathbf{x}_{\mathcal{A}}) = (1 - g) + g = 1$.

The objective function. Finally, we can define the objective function $F : \mathcal{V} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ for the active detection problem, by summing up weights covered from all the edges in $\mathcal{E}$:

$$f_{\text{DET}}(\mathbf{x}_{\mathcal{A}}) = \sum_{(\sigma,v)\in\mathcal{E}} f_{\sigma,v}(\mathbf{x}_{\mathcal{A}}) \tag{5.2.4}$$

The goal of active detection, therefore, is to adaptively select a minimum subset of candidates, such that the edges in the underlying bipartite graph can be fully covered.

5.2.3 ActDet: A Greedy Solution

In this section, we show that the active detection problem defined in the previous section is an *adaptive submodular* optimization problem, and thus can be efficiently solved using a greedy algorithm. First, we show that the objective function (Eq. (5.2.4)) satisfies submodularity:

Lemma 5.2. *f_{DET} is monotone (point-wise) submodular.*

The key idea of the proof is that we can decompose a voting element into many voting elements, each casts equal votes to its favorable candidates. Then we just need to prove F in the new evidence space to be submodular, which is straightforward. We defer the proof details to §A.3.1.

In the active detection setting, we have no access to the label of a candidate in advance. Hence, we are not able to select (candidate, label) *pairs* for each iteration. The following result implies that our objective function also satisfies adaptive submodularity:

Lemma 5.3. *f_{DET} is adaptive submodular w.r.t. $\mathbb{P}[\mathbf{X}_{\mathcal{V}}]$ as long as $X_1, \ldots, X_t$ are independent.*

Proof. With a factorial distribution over the outcomes, the adaptive submodularity of F follows immediately from Lemma 5.2 and Theorem 6.1 of [GK11b]. □

With the objective function defined in §6.1.2, we can associate the following greedy algorithm: It starts with the empty set, and at each iteration adds to the current set $\mathcal{A}$ the candidate v which maximizes the marginal improvement:

$$\Delta_{\text{DET}}(v \mid \mathbf{x}_{\mathcal{A}}) = \mathbb{E}_{\mathbf{x}_v}[f_{\text{DET}}(\mathbf{x}_{\mathcal{A}} \cup \{(v, x_v)\}) - f_{\text{DET}}(\mathbf{x}_{\mathcal{A}}) \mid \mathbf{x}_{\mathcal{A}}] .$$

Algorithm 5: ActDet: The active detection algorithm

Input: Bipartite graph $\mathcal{G}(\Sigma, \mathcal{V}, \mathcal{E})$, prior $\mathbb{P}[\mathbf{x}_{\mathcal{V}}]$, discount factor γ, maximal number of detections to be proposed N

Output: Detections (with associated labels) $\mathbf{x}_{\mathcal{A}}$

begin

1 $\quad \mathcal{A} \leftarrow \emptyset, \quad \mathbf{x}_{\mathcal{A}} \leftarrow \emptyset;$

$\quad$ **foreach** $i \in \{1, \ldots, N\}$ **do**

$\quad\quad$ **foreach** v ***in*** $\mathcal{V}$ **do**

2 $\quad\quad\quad \Delta_+(v) \leftarrow \sum_{\sigma \in \Sigma} \sum_{v' \in \mathcal{V}} \min\{w_{\sigma,v},\ w_{\sigma,v'}\};$

3 $\quad\quad\quad \Delta_-(v) \leftarrow \sum_{\mathtt{cls}(\sigma',v')=\mathtt{cls}(\sigma,v)} \{\gamma \cdot w_{\sigma',v'}\};$

$\quad\quad$ **end**

4 $\quad\quad v^* \leftarrow \arg\max_v \{\mathbb{P}[x_v = +1]\, \Delta_+(v) + \mathbb{P}[x_v = -1]\, \Delta_-(v)\};$

5 $\quad\quad$ Observe x_{v^*};

$\quad\quad$ **foreach** *edges* $(\sigma, v) \in \mathcal{E}$ **do**

$\quad\quad\quad$ **if** $x_{v^*} = +1$ **then**

6 $\quad\quad\quad\quad w_{\sigma,v} \leftarrow \max\{w_{\sigma,v} - w_{\sigma,v^*},\ 0\};$

$\quad\quad\quad$ **else**

$\quad\quad\quad\quad$ **if** $\mathtt{cls}(\sigma, v) = \mathtt{cls}(\sigma, v^*)$ **then**

7 $\quad\quad\quad\quad\quad w_{\sigma,v} \leftarrow \gamma w_{\sigma,v};$

$\quad\quad\quad\quad$ **end**

$\quad\quad\quad$ **end**

$\quad\quad$ **end**

$\quad$ **end**

end

Once the label of v is observed, we update the bipartite graph $\mathcal{G}$ with the remaining edges that have not yet been explained by the current observations $\mathbf{x}_{\mathcal{A}}$. Algorithm 5 provides the details of the greedy algorithm.

Following the analysis of Golovin and Krause [GK11b], we obtain the following performance guarantee:

Corollary 5.4. *Suppose* $f_{\text{DET}} : \mathcal{V} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ *is defined as Eq. (5.2.4). Fix any value* $Q > 0$ *and* $\beta > 0$*, and let* $\text{cost}_{wc}(\pi^*)$ *be the worst-case cost of an optimal policy that achieves a maximum coverage value of* Q *for any possible realization of the variables* $\mathbf{x}_{\mathcal{V}}$*. Let*

$\text{cost}_{wc}(\pi^g_{\text{ACTDET}})$ *be the worst-case cost of Algorithm 5 using a factorial prior on variables* $X_1, \ldots, X_t$*, until it achieves expected value* $Q - \beta$*. Then,*

$$\text{cost}_{wc}(\pi^g_{\text{ACTDET}}) \leq \text{cost}_{wc}(\pi^*) \left(\ln \left(\frac{Q}{\beta} \right) + 1 \right).$$

Moreover, it holds that under the algorithm's prior: $\mathbb{P}\left[f(\mathbf{x}_{\mathcal{A}}) \geq Q \right] \geq 1 - \beta$.

Note that the above result provides guarantees even for worst-case realization of the labels $\mathbf{x}_{\mathcal{V}}$ (i.e., without assumptions on $\mathbb{P}\left[\mathbf{x}_{\mathcal{V}}\right]$), as long as our algorithm uses any factorial prior. Further note that if we choose, in the extreme case, $\beta = \min_{\mathbf{x}_{\mathcal{V}}} \mathbb{P}\left[\mathbf{x}_{\mathcal{V}}\right]$, we guarantee that the greedy algorithm achieves full coverage (i.e., $f(\mathbf{x}_{\mathcal{A}}) \geq Q$) for *all* realizations $\mathbf{x}_{\mathcal{V}}$ and $\mathbf{x}_{\mathcal{A}} \preceq \mathbf{x}_{\mathcal{V}}$. If we do not have a strong prior, we can obtain the strongest guarantees if we choose a distribution that is "as uniform as possible" (i.e., maximizes $\min_{\mathbf{x}_{\mathcal{V}}} \mathbb{P}\left[\mathbf{x}_{\mathcal{V}}\right]$), while still guaranteeing adaptive submodularity.

5.3 Experimental Results

In this section, we empirically evaluate our active detection approach on three (substantially different) datasets: an orangutan nest detection task for biodiversity monitoring, a pedestrian tracking task in a video sequence, and a standard object detection task for the PASCAL VOC Challenge. For each dataset, we employ different base detector that is most tailored to the task. Our emphasis is on comparing the active detection algorithm with classical passive detection, as well as empirically quantifying the improvement by the active detection framework over the base detectors.

5.3.1 Orangutan Nest Detection on UAV-recorded Forest Images

Dataset and experimental setup. The first application is an interactive orangutan nests detection system for biodiversity monitoring. To estimate the distribution of critically endangered Sumatran orangutans (*Pongo abelii*), ecologists deploy conservation drones above orangutan habitat in surveyed areas, so that they can obtain timely and high-quality photographs of orangutan nests high in the tree canopies [KW12]. Our test set contains 37 full-resolution (4000×3000 pixels) images from two separate drone missions launched in September 2012, in Sumatra, Indonesia. Each of the target images

contains at least one orangutan nest, and there are a total number of 45 nests in the dataset, with a minimum size of 19×19 pixels. Selected examples of the nest and non-nest image patches are shown in Fig. 5.5.

(a) Positive examples of orangutan nests

(b) Negative examples

Figure 5.5: Example image patches in the UAV-forest dataset.

As we can see from the above examples, the positive class has high intra-class variation. For efficiency considerations, we reduce the resolution of the original images by half. We then extract all 45 examples of orangutan nests of size 9×9 pixels, as well as 148 background image patches, as the labeled set. Each training example is represented as a 9-dimensional vector which consists of statistics (mean, maximum and minimum) of three color channels in a patch. Based on these features, we train a linear discriminant classifier (LDA) to classify orangutan nests vs. background.

Base detector. The base detector we employ is a sliding-window based system. As we don't have sufficient (positive) training data, we use all the labeled images other than those in the current test image as the training set. At run time, each image patch located by the current sliding window (of size 9×9) is evaluated with a pre-trained classifier and used as a voting element that casts equal votes to its surrounding area (i.e., 9×9 pixels). The confidence of votes from theses voting windows are determined by their distances to the classifier's decision boundary; positive windows that are further away from the decision boundary have higher confidence when voting for a nest candidate.

Clustering the votes. To cluster similar voting elements, we use k-means clustering on the set of voting windows. Moreover, as negative detections often occur adjacently (e.g., branches are usually connected), we also use a *local clustering* algorithm (i.e., segmenting nearby regions), to avoid overwhelming false detections. As an example,

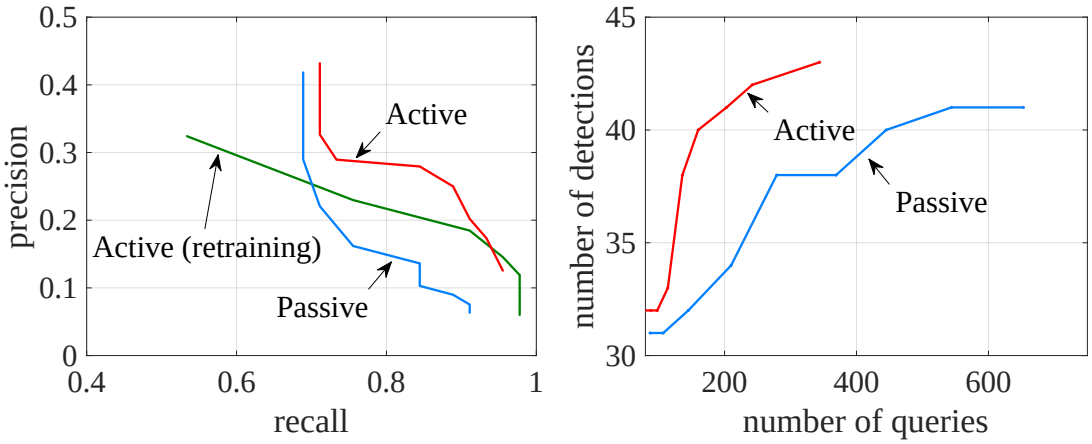


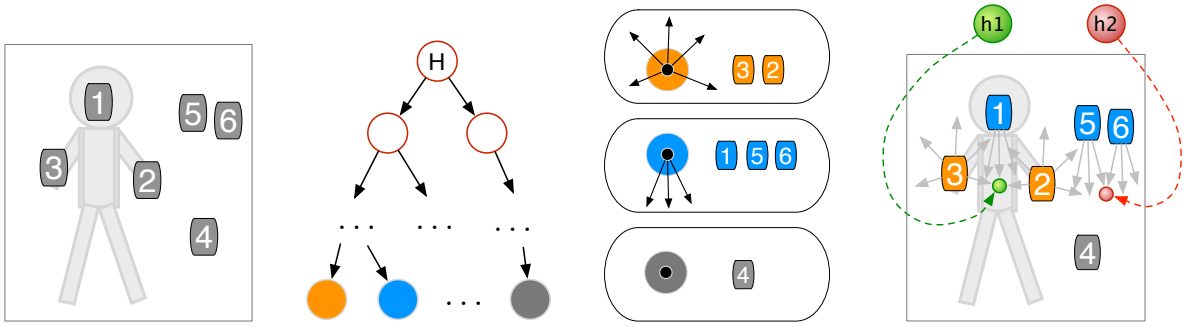
Figure 5.6: Results: Performance of the active detection on UAV-FOREST.

see the response image demonstrated at Fig. 5.1. Once we query some candidate at the "Y" branch and get to know its label, we will either cover (when the detection is true), or discount (when the detection is false) the weights for the entire local region. In our experiments, we use mean-shift [CM02] for local clustering. We adaptive the code from the EDISON [Lab02] system, and set the parameters as follows: a spatial width $h_s = 25$, a feature (range) width $h_r = 10$, and a minimum region area $M = 169$. We adjust the spatial width so as to avoid discounting too many regions by a single observation while being able to maintain a relatively low computational cost.

Results. We study two stopping criteria to decide when to quit the detection process. The first one, which is used to generate the results in Fig. 5.6, is to stop when the marginal gain by selecting a candidate falls below some particular value. The second choice is to stop when the accumulated coverage of the votes exceeds a certain fraction of the total votes in the images. While the first one seems to be a practical choice, the second is in direct connection with the adaptive minimum-cost coverage problem, where one stops once the total utility exceeds a fixed quota. From Fig. 5.6, we can see that active detection significantly outperforms the passive algorithm: At 80% recall, active detection ($\gamma = 0.5$) obtains almost twice the precision (0.27 vs. 0.15) as the passive algorithm.

5.3.2 Pedestrian Detection on TUD-crossing Sequence

Base detector. Hough based approaches offer seamless integration with the active detection framework. To demonstrate how user supervision can help with such systems, we apply Algorithm 5 to the TUD-crossing sequence, based on the *Hough Forest* detector proposed in by Gall and Lempitsky [GL09]. Essentially, they produce a set of individual voting elements that are integrated to reason about the existence of a full object. Concretely, a direct mapping between the appearance of an image patch and its Hough votes is learned through a random forest framework, as illustrated in Fig. 5.7.



(a) *Input image* (b) *Hough forest* (c) *Clusters* (d) *Hough votes & detection*

Figure 5.7: Illustration of the Hough-forest detection algorithm.

Each tree in the forest is constructed based on a set of patches that are sampled from the training collection of images (Fig. 5.7a), and each leaf node in the tree stores the proportion of image patches that belong to an object, and their corresponding offset vectors from the object centroid (Fig. 5.7b). At runtime, patches of the test image are fed to the forest and passed through the branches of the Hough trees. Then the information stored at the leaf nodes of the trees is used to cast probabilistic votes about the existence of the object at some location (Fig. 5.7c, Fig. 5.7d).

Fig. 5.8 shows a frame from the sequence and its corresponding response image.

Clustering the edges. Other than the direct benefit of a ready-to-use voting scheme, the Hough forest framework also provides a natural and intrinsic characterization of the "similarity" of votes. If two image patches fall into the same leaf node, then by default they are clustered together (see Fig. 5.7c). Therefore, when receiving negative feedback from an external expert, the active detector can efficiently update the bipartite

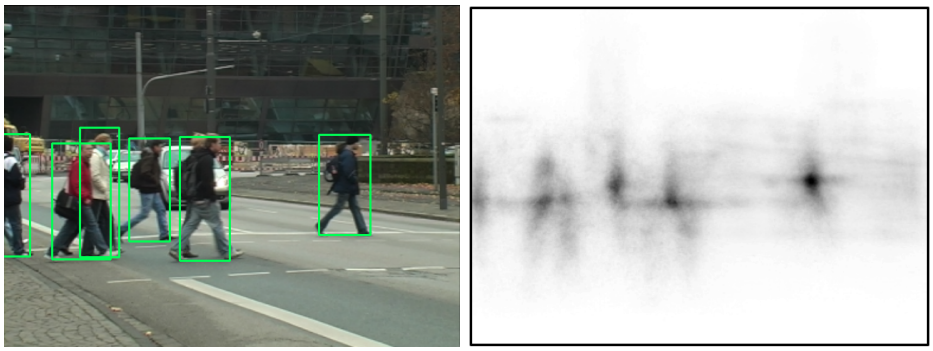


Figure 5.8: Pedestrian detection (TUD-CROSSING).

graph, by adjusting the weight (i.e., probabilities) of similar votes that share the same leaves for all the trees in the forest. Note that we are clustering edges in the bipartite graph, not voting elements. Therefore, the updates should be focused only on the votes that are pointing to the same directions as the false votes. As an example, in Fig. 5.7d, if the active detector proposes h_2, and finds it to be a false detection, then we will discount the weights for (1) the rightmost edge for voting element v_5, and (2) the leftmost edge for voting element v_6. In other words, we just need to discount the two corresponding edges in the blue cluster (Fig. 5.7c) – such that all blue voting elements are updated.

We use a discount factor $\gamma = 0.01$ to penalize votes that are "similar" with any of the incorrect votes. We also use "local clusters" to update the bipartite graph when observing a false candidate, similar to the case for the nest detection task: edges that share the same voting element are considered within the same local cluster, and thus will be discounted if any of them points to a false candidate.

Since the background clutter doesn't change much across frames, for active detection we choose to share the cluster updates through the entire video sequence, rather than discard the information acquired from user feedback and start from scratch (i.e., reset the negative count for each cluster) for each new frame.

Matching bounding boxes. Once the set of candidate bounding boxes are proposed, we need to match them with the ground truth bounding boxes for evaluation. Note that greedy matching is problematic for datasets that exhibit sufficient overlap between objects because once a detection is matched to one object, it cannot switch to another object even if the second is a better matching. Instead, we apply Hungarian algorithm

to match the set of detections with the ground truth annotations, based on the Jaccard similarity (similarity $\leq$ 40% are considered as false detections) between bounding boxes.

Results. We compare active detection with the state-of-the-art passive detection results on this dataset, which is given by Barinova, Lempitsky, and Kholi [BLK12a]. We test the candidate algorithms on 41 frames of the TUD-CROSSING sequence (by sampling every 5th frame of the full video sequence) in the single scale scenario, and show the results in Fig. 5.12. The curves are generated by varying the stopping threshold on the margin gains of new candidates. We limited the maximum number of detections to be 10 for both systems (given there are at most eight objects per frame) to have a fair comparison. As can be seen, with user supervision, our framework considerably outperforms the baseline detection algorithm.

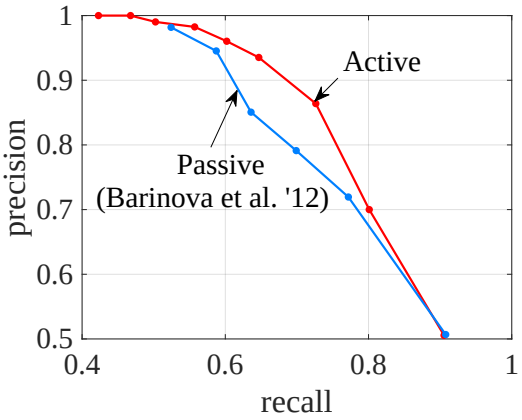
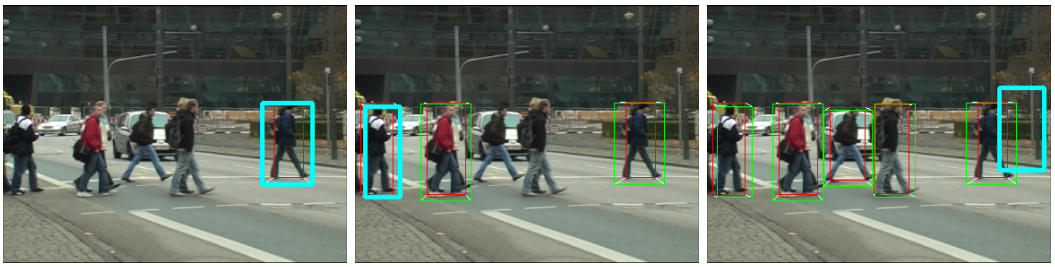


Figure 5.12: Results: TUD-CROSSING.

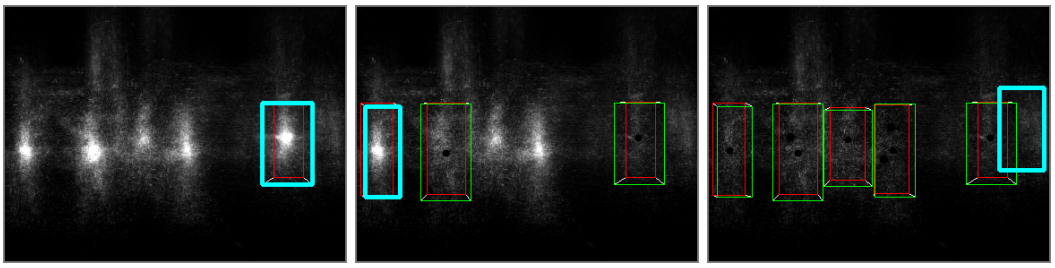
Fig. 5.9, Fig. 5.10, and Fig. 5.11 demonstrate the dynamics of the active detection process. The current detection is highlighted by cyan bounding box. Red bounding boxes indicate the ground truth labels of pedestrians, and green bounding boxes are the detections made by ACTDET (Alg. 5). For illustration purpose, we only show the bounding boxes of the true detections, and that of the current detection (regardless of its label).

We start active detection from the 16th frame of the video sequence (Fig. 5.9a). Each time we finish detecting one frame, we take the next frame that is 5 frames away from the current one as input. That is, the sequence of test images are frame 16, 21, 26, 31, 36, 41, 46, 51, etc.

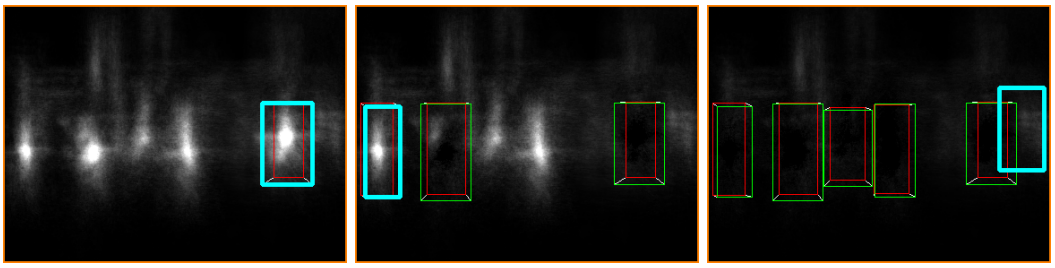
In Fig. 5.9, we show three detections made by the active detector. As we can see from the third row of Fig. 5.9a, the active detector makes a false prediction at the 9th iteration, where it mistakes the pole as a pedestrian. According to the negative update rule in the active detection framework, we will then discount all the similar votes, which indicates that, in the following frames, the negative coverage by selecting a similar candidate (e.g., detecting the pole again) will be discounted.



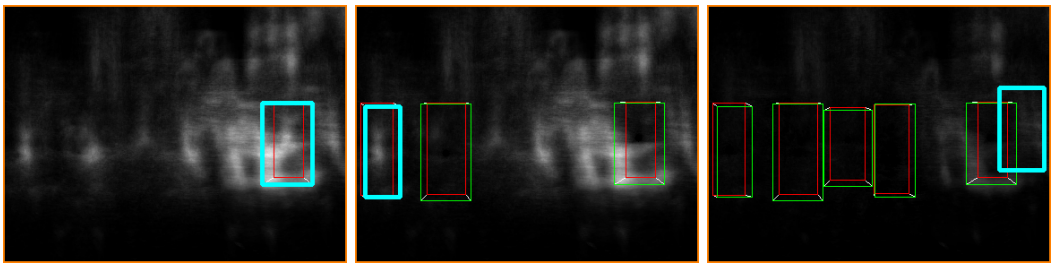
(a) *Input image*



(b) *Response image*



(c) *Positive coverage*



(d) *Negative coverage*

Figure 5.9: The active detection results on the 16th frame of the TUD-CROSSING sequence, at the 1st, 3rd, and 9th iteration. Each column illustrates the dynamics of the corresponding items. By "positive (resp. negative) coverage", we mean the total sum of edge weights covered by an observation a positive (resp. negative)label at given locations.

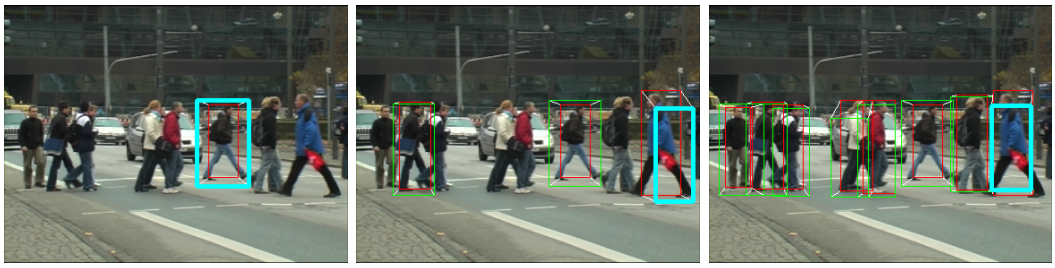
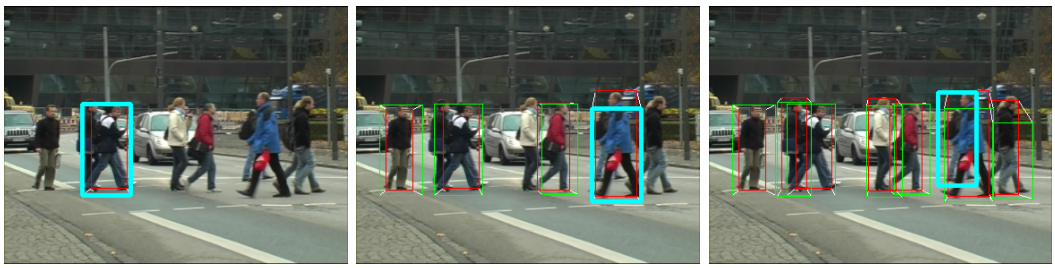


Figure 5.10: The active detection results on the 41st frame of the TUD-CROSSING sequence, at the 1st (first row), 3rd (second row), and 9th iteration (third row).



(a) *Input image*

Figure 5.11: The active detection results on the 51st frame of the TUD-CROSSING sequence, at the 1st (first row), 4th (second row), and 8th iteration (third row).

Fig. 5.10 demonstrates the above changes. Similar to Fig. 5.9, we show three detections made by the active detector, on the 41st frame of the video sequence (i.e., the 6th test image). The active detector managed to ignore the pole as a candidate detection up to the 9th iteration. Another interesting result we found in Fig. 5.10 is, by using the Hungarian matching algorithm, we can actually switch the associated ground truth object of the detections made in the previous iterations, to a better matching. For example, in the second row and third row of Fig. 5.10, the algorithm proposes two detections (illustrated as cyan bounding boxes), both could be matched to the same pedestrian. When the dynamic matching algorithm finds a better matching at the 9th iteration (third row), it discards the predictions made at the 3rd iteration (second row), and therefore gives better detection results.

Furthermore, we show in Fig. 5.11 how our active detection framework can detect highly overlapped objects, without invoking the non-maximum suppression mechanism. In the first row, it starts to detect the 8th input image (i.e., the 51st frame). When it find the pedestrian (in blue jacket) closer to the camera at the 4th iteration (second row), the algorithm doesn't remove the entire surrounding region. Instead, it only covers its

direct votes and all neighboring votes. Those edges that are not in any form associated with the candidate that represent the pedestrian detected are not affected. As a result, in the third row (8^{th} iteration), the algorithm successfully identifies the pedestrian further from the scene – which is nearly 90% occluded. In this scenario (i.e., dealing with highly overlapped objects), our algorithm behaves in a similar manner as that of [BLK12a], but it has the advantage that, it allows us to address the active detection problem – by invoking the adaptive submodularity property of the objective function – in a principled way.

5.3.3 Object Detection on PASCAL VOC Dataset

Dataset and base detector. The third data set differs from the previous two in the sense that it contains object classes that exhibit much richer structural features (e.g., the "person" class includes examples of a high variability of poses and shapes). The state-of-the-art results for this dataset are obtained by the sliding-window based, multi-scale, deformable parts model (MDPM) of [Fel+10]. To convey the idea that our framework can incorporate different base detectors, we build our bipartite graph upon an earlier release (*voc-release3*) of their system, as it already includes most of the important innovations of the MDPM, without additional expensive components (e.g., grammar models as described in [GFM11]) that are designed specifically for certain tasks.

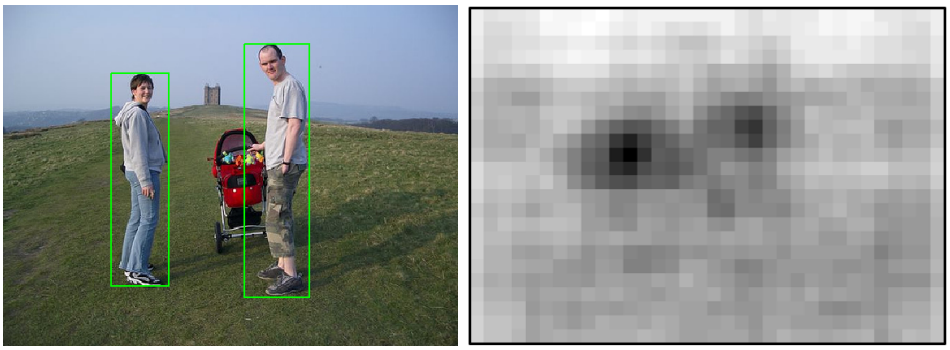


Figure 5.13: Person detection (PASCAL VOC 2008).

In MDPM, each category is modeled by a "root" filter that describes the overall shape of the object, and a fixed number of "part" filters that describe important sub-areas of the object at a higher resolution. For multi-scale detection, we keep a *feature pyramid* that consists image *cells* (of size 8×8, represented by a 31-dimensional HOG

descriptor [DT05]) from a pile of rescaled versions of the image. A candidate v is then characterized by the location and scale of an object and is scored jointly by both root filter and associated parts filters.

Constructing the voting graph. To build a bipartite graph, we assume that voting elements correspond to image patches (i.e., cells in the feature pyramid), and will cast equal votes for a candidate v given that they are inside its associated window. The total sum of votes v receives from the voting cells amount to the score provided by the underlying MDPM. To handle the deformable parts, we further assume two types of candidates: "root candidates" that represent the existence of an object, and "part candidates" as intermediate nodes in the bipartite graph, that can be voted by (part) cells. Each candidate node in the bipartite graph will eventually receive (direct) votes from the root cells, as well as (indirect) votes from the part cells, that are weighted by the deformation coefficient [Fel+10] of the part window.

Edge similarity is measured based on two sets of features: the filter type of the window associated with candidate v, and the HOG descriptor of the voting cell σ. To construct the clusters, we first group the windows (roots and parts separately) by filter type, and then employ a hierarchical clustering method to retrieve similar edges (i.e., small cosine distance between HOG descriptors).

We use the same evaluation criterion as provided by PASCAL VOC Challenge. Fig. 5.14 shows our results on the *Person* category of the *VOC2008* data set. Note that the active detector (AUC 0.566) as tested in this experiment only uses root filters, yet it already outperforms the passive baseline system (AUC 0.516) that utilizes both root and parts filters. We find that the passive approach (i.e., "active detection" assuming all detections are true) also considerably outperforms the baseline (AUC 0.544). One possible reason is that when identifying multiple objects, our framework by default bypasses the non-maximum suppression approach and thus has better recall.

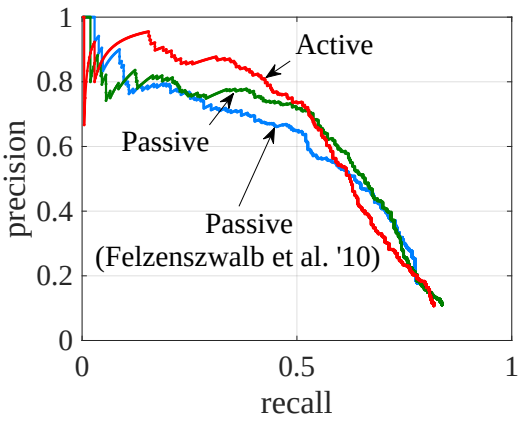


Figure 5.14: Results: PASCAL VOC 2008 "person" category.

5.4 Related Work on Object Detection

Multiple object detection via submodular optimization. Sliding-window based algorithms [Fel+10] and patch based (e.g., Hough transform based) algorithms are two of the most popular approaches for multiple object detection. These methods produce responses with peaks at candidate object locations. When dealing with overlapping hypotheses, conventional object detection methods use non-maximum suppression or mode-seeking to locate and distinguish peaks. Such post-processing requires tuning of many parameters and is often fragile, especially when objects are located spatially close to each other. Alternatively, Barinova, Lempitsky, and Kholi [BLK12a] propose a probabilistic framework for multiple object detection, with an objective function satisfying submodularity, which can be solved efficiently with a greedy algorithm for submodular maximization [NWF78]. Hereby, submodularity captures diminishing returns in the detector response at nearby object locations. In their framework, image patches are employed as *voting elements*, which can cast probabilistic votes for *hypotheses* that encode some specific configuration of an object. Our work is inspired by this framework. In contrast to their approach, however, we consider the active detection setting, where detection is interleaved with expert feedback.

Active learning for object recognition. Active learning has been used successfully for reducing labeling cost in a classification setting [JPP12], given a large amount of unlabeled data. However, it has been shown to be more challenging for object detection problems. These approaches start off with few annotated images and then look at a pool of unlabeled examples, and find the ones which would most improve the performance of the classifier once their label has been obtained. This procedure has been shown to significantly reduce the number of required labels [AF04; Kap+07; Bie12]. Vijayanarasimhan and Grauman [VG11] employ active learning together with crowdsourcing techniques on web-crawled images to provide an end-to-end system for automatically learning object detectors. In the existing literature, however, active learning has only been applied to train a good base detector. In this work, we consider the complementary setting, of taking a base detector (trained, e.g., using active learning), and applying it in an active manner, i.e., interleaving automatic detection with expert feedback.

5.5 Summary

We proposed an active detection framework that enables turning existing base detectors into automatic systems for intelligently interacting with users. Our approach reduces active object detection to a sequential edge covering optimization problem. Our objective function satisfies adaptive submodularity, allowing us to use efficient greedy algorithms, with strong theoretical performance guarantees. We demonstrated the effectiveness of the active detection algorithm on three different real-world object detection tasks, where showed that active detection not only works for various base detectors but also provides substantial advantages over its passive counterpart.

Part III

Practical Challenges

6

Exploiting Information Parallelism

Thus far, we have considered adaptive information acquisition as a *sequential* decision process. However, in many practical implementations (such as crowdsourcing, surveys, high-throughput experimental design), it is preferable to acquire information *in parallel* due to time and resource constraints, for example, in situations where tests performed in parallel could have shared cost. Such information gathering setting have been studied from the perspective of *batch-mode* active learning. While several heuristics have been proposed for such problems, little is known about their theoretical performance.

In this chapter, we consider information-parallel stochastic optimization problems that exhibit *adaptive submodularity*. We prove that for such problems, a simple greedy strategy is competitive with an optimal batch-mode policy. In some cases, perhaps surprisingly, the use of batches incurs competitively low cost, even when compared to a fully sequential strategy. We demonstrate the effectiveness of our approach on batch-mode active learning tasks, where it outperforms the state of the art, as well as the novel problem of multi-stage influence maximization in social networks.

Organization of this chapter. We start by stating the general batch-mode stochastic optimization problem in §6.1. In §6.2, we focus on the extreme case where all information are acquired in a single batch, and hence the batch-mode information acquisition problem reduces to a non-adaptive optimization problem. In §6.3, we consider the more general batch-adaptive setting, and prove strong performance guarantees for a simple

greedy algorithm. We provide practical algorithms for batch-mode active learning and influence maximization problem in §6.4, and demonstrate the empirical effectiveness of the algorithms for both applications in §6.5. In §6.6 we summarize the chapter.

6.1 Problem Statement

We first describe two different applications that fall into the batch-mode setting. Then, in Section 6.1.2 we introduce a formalism that captures both problems.

6.1.1 Motivating applications

Example 6.1 (Multi-stage influence maximization in social networks)**.** *Suppose we would like to stimulate demand for a new product. By giving the product to a subset of target people for free, we hope these people can influence their friends, potentially creating a cascade of influence motivating many more consumers to adopt the product. Instead of committing to all target nodes in advance, it is natural to consider conducting a* multi-stage *advertising campaign: In each stage, a subset of k nodes are targeted, then the effect of the campaign is observed, then the next k target nodes are chosen, and so on. Implementing such a procedure may be much more practical if many (i.e., for $k > 1$) nodes can be selected in each stage, to be influenced in parallel. Our goal is to find the best strategy to select (batches) of people to target, so that influence of the selected subset can be maximized.*

The above problem was formalized by Kempe, Kleinberg, and Tardos [KKT03] as a stochastic optimization problem, who show that many natural diffusion models (such as the independent cascade [GLM01], or linear threshold models [Gra78]) can fit into a general stochastic optimization framework based on submodular functions. Their results were later generalized to the adaptive setting in [GK11a]. As a concrete example, let us consider the independent cascade model. Formally, let $\mathcal{V} = \{1, \ldots, t\}$ be the set of nodes in the social network, and let X_v be the (random) set of nodes eventually influenced if v is initially targeted (X_v denotes the outcome of "test" v if we stick to the terminology introduced in §2.1.1). In the independence cascade model, we assume a factorial distribution over the test outcomes. Therefore, if a set $\mathcal{A}$ of nodes is initially targeted, the eventual influence is $\bigcup_{v \in \mathcal{A}} X_v$ with probability $P(\mathbf{X}_\mathcal{A}) = \prod_{v \in \mathcal{A}} P(X_v)$.

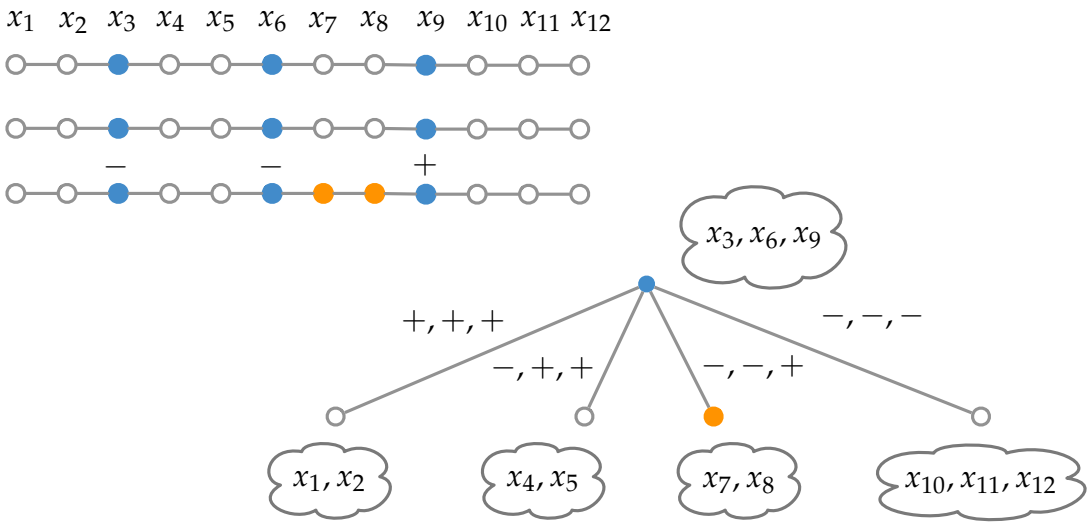


Figure 6.1: Illustration of batch mode active learning (with batch size $k = 3$), in the simple case of one-dimensional data and binary threshold hypotheses. The upper-left figure shows the unlabeled data (top row), the first batch selected for labeling (middle row), and received labels, as well as second selected batch (bottom row). The lower-right figure illustrates the decision tree representing a batch-adaptive policy.

Example 6.2 (Batch-mode pool-based Bayesian active learning (BMAL))**.** *We are given a pool of unlabeled examples indexed by* $\mathcal{V} = \{1, \ldots, t\}$*, with (initially unknown) labels* $x_1, \ldots, x_n \in \{+1, -1\}$*. We wish to learn a classifier* $h : \mathcal{V} \to \{+1, -1\}$ *out of a finite set* $\mathcal{H}$ *of hypotheses, each corresponding to distinct labelings of the pool* $\mathcal{V}$*, and containing the true labeling, i.e., a hypothesis* h *such that* $h(v) = x_v$ *for* $v \in \mathcal{V}$*. An optimal active learning strategy is one that minimizes the expected number of labels requested, in expectation over a given prior* $\mathbb{P}[H]$*. We consider strategies for* batch-mode *active learning, which pick* batches *of* k *unlabeled examples at a time, then request all labels for the selected batch in parallel, and then proceed to pick the next batch given the labels obtained so far.*

As discussed in Part II of this dissertation, the pool-based active learning problem has been extensively studied under the sequential setting, a classical solution being GBS. In essence, GBS attempts to selects the examples that can shrink the probability mass of the version space as quickly as possible (recall from §4.1.4 that we use $\mathcal{H}(\mathbf{x}_\mathcal{A}) = \{h \in \mathcal{H} : i \in \mathcal{A} \Rightarrow x_v = h(v)\}$ to refer to the version space consistent with the observation $\mathbf{x}_\mathcal{A}$.). Analogously, under the batch-mode setting, we wish to select *batches of examples* of size k that can quickly shrink the version space, such that these examples allow us to uniquely identify h (i.e., to infer the labels of all unlabeled examples, or equivalently

to have $|\mathcal{H}(\mathbf{x}_\mathcal{A})| = 1$). See Figure 6.1 for an illustration of the batch-mode policy for learning 1-dimensional threshold function under the realizable setting.

6.1.2 General Problem Statement

Under the sequential setting, both applications of §6.1.1 can be captured by the general *adaptive information acquisition* problem as stated in §2.1: Given the joint probability distribution $\mathbb{P}[\mathbf{X}_\mathcal{V}]$ and reward $f(\mathbf{x}_\mathcal{A})$ of observations $\mathbf{x}_\mathcal{A}$, we want to find the best policy for selecting tests. These applications differ by their choices of reward function f. For the influence maximization example, we can use $f(\mathbf{x}_\mathcal{A}) = |\bigcup_{v\in\mathcal{A}} x_v|$, i.e., the number of nodes eventually influenced. For our active learning example, we can use $f(\mathbf{x}_\mathcal{A}) = |\mathcal{H}| - |\mathcal{H}(\mathbf{x}_\mathcal{A})|$, i.e., the number of the hypotheses eliminated through the labeled examples $\mathbf{x}_\mathcal{A}$.

In both applications, f is *adaptive monotone*, *adaptive submodular* (§2.3.2), and *pointwise submodular* (§2.3.4). In this chapter, we will mainly discuss the batch-mode extension of the *adaptive minimum cost coverage problem* (Problem 2.1.3). That is, we aim to find a policy π for selecting items (and associated observations) $\mathbf{x}_\mathcal{A}$, such that we achieve a certain quota of value $Q \geq 0$, i.e., $f(\mathbf{x}_\mathcal{A}) \geq Q$, while at the same time minimizing the cost of performing the tests in $\mathcal{A}$. In the influence maximization example, Q may be a certain fraction of the size of the social network. In active learning, $Q = |\mathcal{H}| - 1$: achieving this quota is a necessary and sufficient condition for identifying the true hypothesis. In the following, w.l.o.g.[1], we assume $f(\mathbf{x}_\mathcal{V}) = Q$ for all $\mathbf{x}_\mathcal{V} \in \mathrm{supp}(\mathbb{P}[\mathbf{X}_\mathcal{V}])$.

Unless explicitly pointed out, we assume *unit cost* for each test, i.e., $c(v) = 1$ and $C(\mathbf{x}_\mathcal{A}) = \sum_{v\in\mathcal{A}} c(v) = |\mathcal{A}|$. Therefore, our definition of the expected and worst-case cost of policy π in §2.1.2 reduce to

$$\mathrm{cost}_{avg}(\pi) = \mathbb{E}_{\mathbf{x}_\mathcal{V}}[|\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})|]\,;\ \mathrm{cost}_{wc}(\pi) = \max_{\mathbf{x}_\mathcal{V}} |\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})|$$

where recall $\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})$ represents the set of tests that π selects, given that the outcomes of the realized tests are consistent $\mathbf{x}_\mathcal{V}$. Now let Π denote the set of candidate policies from which we can choose. This could be, for example, all batch-mode policies of batch size k. We want to find a feasible policy π^* with minimum cost,

$$\pi^* \in \arg\min_{\pi\in\Pi} \mathrm{cost}(\pi),\ \text{subject to } f(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) \geq Q \text{ for all } \mathbf{x}_\mathcal{V} \text{ with } \mathbb{P}[\mathbf{x}_\mathcal{V}] > 0.$$

[1] If $f(\mathbf{x}_\mathcal{V}) > Q$, we replace f with the (submodular) function $f_Q(\mathbf{x}_\mathcal{A}) = \min(f(\mathbf{x}_\mathcal{A}), Q)$.

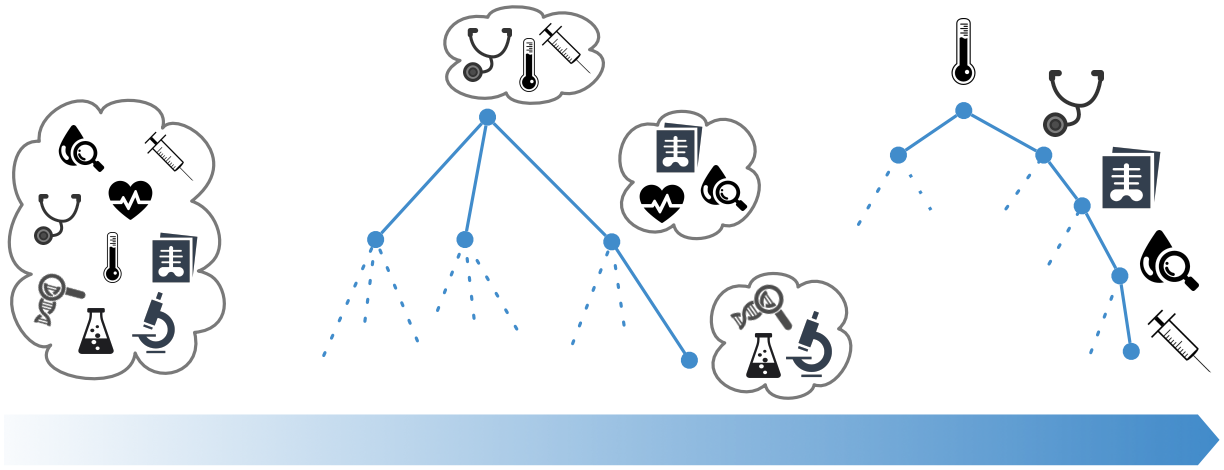


Figure 6.2: Information Acquisition: from full-batch (non-adaptive) setting to sequential.

Batch selection Based on the above notations, we can study how different classes of policies compare in terms of their cost. Denote the set of policies by Π. On the one extreme, we have *fully sequential* policies, denoted by $\Pi_{\text{seq}} (\subset \Pi)$, where the choice of each item may depend on the labels of *all* previous items selected. On the other extreme, we have *constant*, or non-adaptive policies $\Pi_{const} (\subset \Pi)$ which commit to items picked in advance, before making any observations (and therefore it does not depend on the observations). However, fully sequential and constant policies are only two extremes on a spectrum (See Fig. 6.2). We are interested in policies $\Pi_{[k]}$ that sequentially pick *batches* of size k. Any policy $\pi \in \Pi_{[k]}$ starts selecting a fixed set $\mathcal{A}_1 \subseteq \mathcal{V}$ of k items. It then obtains all labels $\mathbf{x}_{\mathcal{A}_1}$. If $f(\mathbf{x}_{\mathcal{A}_1}) \geq Q$, it stops. Otherwise, if batches $\mathcal{A}_1, \ldots, \mathcal{A}_{\ell-1}$ have already been selected, it picks batch $\mathcal{A}_\ell \subseteq \mathcal{V}$ of k items, obtains the labels, and stops if $f(\mathbf{x}_{\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_\ell}) \geq Q$.

Obtaining an optimal batch policy is a formidable task. In fact, even *representing* an optimal batch policy may require exponential space: There are $\binom{n}{k}$ batches of size k, and an optimal batch policy assembles such batches into a decision tree of possibly exponentially large branching factor. In the following sections, we will study how to design an efficient batch policy, while guaranteeing that it is competitive with the optimal policy. In particular, we describe a simple greedy algorithm and prove that it implements a batch policy with cost competitive to that of the optimal batch policy. Moreover, we prove that under some additional conditions on the distribution $\mathbb{P}[\mathbf{X}_\mathcal{V}]$, the greedy algorithm is even competitive with the optimal *fully sequential* policy.

6.2 Open-loop Information Acquisition

Before we analyze the behavior of a batch-adaptive policy, we first dive into the non-adaptive setting, and take a look at how a (constant) greedy policy performs within a *single* batch, compared with the optimal *sequential* policy. A key notion we use for analyzing the performance of a non-adaptive policy is the *adaptivity gap*, defined as follows.

Definition 6.3 (Adaptivity Gap)**.** The adaptivity gap of $(f, \mathbb{P})$ and cardinality constraint $k (k \leq |\mathcal{V}|)$ is defined as

$$\textsc{gap}_k(f, \mathbb{P}) = \frac{\textsc{opt}_{\text{seq}}(k)}{\textsc{opt}_{const}(k)},$$

where $\textsc{opt}_{\text{seq}}(k) = \max_{\pi \in \Pi_{\text{seq}}} \mathbb{E}_{\mathbf{x}_\mathcal{V}}[f(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V}))]$ denotes the *optimal sequential expected reward* of f under cardinality constraint k, and $\textsc{opt}_{const}(k) = \max_{\pi \in \Pi_{const}} \mathbb{E}_{\mathbf{x}_\mathcal{V}}[f(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V}))]$ denotes the optimal constant expected reward under the same constraint.

From now on we use the notation $\textsc{gap}_k = \sup_{(f,\mathbb{P})} \textsc{gap}_k(f, \mathbb{P})$ and we call it the adaptivity gap under cardinality constraint k.

6.2.1 Factorial Prior Distribution

To begin with, we assume that the variables $X_1, \ldots, X_n$ are *independent*. This assumption is satisfied in the influence maximization application, e.g., when we assume the independent cascade diffusion model. An important result that relates $\textsc{opt}_{\text{seq}}$ with $\textsc{opt}_{const}$ is by Asadpour and Nazerzadeh [AN15], which we state as below.

Theorem 6.4 (Adapted from Asadpour and Nazerzadeh [AN15])**.** *Let $f : 2^{\mathcal{V} \times \mathcal{X}} \to \mathbb{R}_{\geq 0}$ be monotone and submodular, and $\mathbb{P}[\mathbf{X}_\mathcal{V}]$ such that $(f, \mathbb{P})$ is adaptive submodular. Assume the set of variables $\mathbf{X}_\mathcal{V} = \{X_1, \ldots, X_v\}$ are independent, i.e., $\mathbb{P}[\mathbf{X}_\mathcal{V}] = \prod_{v \in \mathcal{V}} \mathbb{P}[X_v]$. Then, for any k, there exists a non-adaptive policy of length k (i.e., running the policy till selecting k items) that approximates the optimal (sequential) adaptive policy of length k within a factor of $1 - 1/e$. In other words, the adaptivity gap*

$$\textsc{gap}_k < \frac{e}{e-1}.$$

Algorithm 6: The NON-ADAPTIVE GREEDY algorithm.

Input: Batch size k, reward function $f : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$ and prior $\mathbb{P}[\mathbf{X}_{\mathcal{V}}]$;
Output: Selected items $\mathcal{A}$. **begin**
  $\mathcal{A} \leftarrow \emptyset$;
  while $|\mathcal{A}| \leq k$ **do**
    $v \leftarrow \arg\max_{v'} \mathbb{E}_{\mathbf{x}_{\mathcal{V}}}[f(\mathcal{A} \cup \{v'\}, \mathbf{x}_{\mathcal{V}}) - f(\mathcal{A}, \mathbf{x}_{\mathcal{V}})]$;
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{v\}$;
  end
end

We refer interested readers to [AN15] for the proof of Theorem 6.4. This result implies that under a factorial prior, the *price of parallelism* is always bounded by a constant, independent of the cardinality constraint k.

Note that finding the optimal non-adaptive policy is also NP-hard. In practice we choose to run a simple greedy algorithm (Algorithm 6). Since we assume f is adaptive monotone and pointwise submodular, we know that the set function $F(\mathcal{A}) = \mathbb{E}_{\mathbf{x}_{\mathcal{V}}}[f(\mathcal{A}, \mathbf{x}_{\mathcal{V}})]$ is monotone submodular. Therefore, by [NWF78] we know Algorithm 6 achieves a $1 - 1/e$ approximation of the optimal non-adaptive policy. Combining with Theorem 6.4 we obtain the following corollary:

Corollary 6.5. *Let $f : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{N}$ be monotone and submodular, and assume $\mathbb{P}[\mathbf{X}_{\mathcal{V}}]$ is a factorial distribution such that $(f, \mathbb{P})$ is adaptive submodular. Denote the greedy (constant) policy induced by Algorithm 6 as π^{g}_{const}, and the optimal sequential policy as π^{*}_{seq}. If we run π^{g}_{const} and π^{*}_{seq} under the same cardinality constraint, then it holds that*

$$F(\pi^{g}_{const}) \geq F(\pi^{*}_{seq}) \cdot \frac{1}{\mathrm{GAP}_k}\left(1 - \frac{1}{e}\right) > \left(1 - \frac{1}{e}\right)^2 F(\pi^{*}_{seq}), \tag{6.2.1}$$

where $F(\pi) := \mathbb{E}_{\mathbf{x}_{\mathcal{V}}}[f(\mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}}))]$ denotes the expected reward of a policy π.

6.2.2 Adaptivity Gap for Arbitrary Distribution[2]

Now we turn our attention to the general case where the distribution $\mathbb{P}$ over test outcomes $\mathbf{X}_{\mathcal{V}}$ is not factorial. Many practical applications fit into this setting. For

[2]Results presented in §6.2.2 are based on unpublished work in collaboration with Gábor Bartók, who provided the proofs of Theorem 6.6 and Theorem 6.7.

example, in active learning, the label of an example depends on other examples; in the influence maximization application, (instead of the independent cascade model) we may assume that the probability one node succeeds in influencing its neighbors depends on its record in previous rounds. In this following, we state the adaptive gap under general distributional assumptions.

We first establish an upper bound on the adaptivity gap for cardinality constraint $k = 2$.

Theorem 6.6. *Let $f : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{R}_{\geq 0}$ be monotone and submodular, and $\mathbb{P}[\mathbf{X}_{\mathcal{V}}]$ such that $(f, \mathbb{P})$ is adaptive submodular. The adaptivity gap for $(f, \mathbb{P})$ and cardinality constraint $k = 2$ can be upper bounded as*

$$\textsc{gap}_2 \leq \frac{e}{e-1}.$$

The proof of Theorem 6.6 relies on a reduction to the factorial distribution setting for $k = 2$, and thus we can use the Theorem 6.4 to obtain the bound. To keep the discussion concise we defer the details of the proof to §A.4.1. Based on this base case, we generalize the adaptivity gap to arbitrary cardinality constraint k:

Theorem 6.7. *For general adaptive submodular functions f and arbitrary distribution $\mathbb{P}$, the adaptivity gap for cardinality k can be upper bounded as*

$$\textsc{gap}_k \leq \frac{2\textsc{gap}_2}{\textsc{gap}_2 - 1} k^{\log(\textsc{gap}_2)}.$$

Moreover, in the special case where k is a power of 2, we can further tighten the bound as

$$\textsc{gap}_k \leq \frac{\textsc{gap}_2}{\textsc{gap}_2 - 1} k^{\log(\textsc{gap}_2)}.$$

The general proof strategy is to first use induction to prove the case where k is a power of 2, and then generalize the result to arbitrary k using adaptive submodularity. The full proof will appear in §A.4.2. Combining Theorem 6.6 and Theorem 6.7 we obtain $\textsc{gap}_k \leq e \cdot k^{0.662}$ for k being a power of 2, and $\textsc{gap}_k \leq 2e \cdot k^{0.662}$ for general $k \leq |\mathcal{V}|$. Note that the bound in Theorem 6.7 is non-trivial whenever $\log(\textsc{gap}_2) < 1$, that is, if $\textsc{gap}_2 < 2$. If $\textsc{gap}_2$ can have a tighter upper bound, the upper bound on the adaptivity gap gets tighter as well.

6.3 BatchGreedy: Greedy Approach and Guarantees

We now investigate the batch-adaptive setting, where we consider a simple, greedy approach towards constructing batch policies. This policy, BatchGreedy, selects

items *within* a batch in a greedy manner (i.e., following Algorithm 6), then receives observations for all items in the batch, then selects the next batch in a greedy manner, conditional on all observations made so far, and so on. An important challenge in batch selection is the fact that the value of items (e.g., unlabeled examples) selected depends on observations (e.g., labels) obtained only *after* the entire batch is selected. In active learning, for example, one wishes to select examples within a batch that are likely to be informative individually, but also diverse (minimize redundancy). BatchGreedy addresses this challenge by using a suitable notion of *marginal benefit* of an item, that takes into account all observations made so far, as well as items that have already been selected within the batch (but no observation has been obtained yet). Formally, we generalize the *conditional marginal benefit* of item v (c.f. Eq. (2.3.2)) by

$$\Delta_f(v \mid \mathcal{B}, \mathbf{x}_\mathcal{A}) = \mathbb{E}_{\mathbf{x}_\mathcal{V}}\left[f(\mathbf{x}_{\{v\}\cup\mathcal{B}\cup\mathcal{A}}) - f(\mathbf{x}_{\mathcal{B}\cup\mathcal{A}}) \mid \mathbf{x}_\mathcal{A}\right]. \tag{6.3.1}$$

Thus, $\Delta_f(v \mid \mathcal{B}, \mathbf{x}_\mathcal{A})$ reflects the expected marginal gain of item v, when items $\mathcal{A}$ have been selected and the corresponding observations $\mathbf{x}_\mathcal{A}$ have been made, and items $\mathcal{B}$ have already been selected, but *no* observations have yet been made about them. Therefore, Eq. (6.3.1) captures possible redundancy (diminishing gains) of candidate item v w.r.t. to labels already obtained, as well as labels that will likely be obtained within the batch. Hence it encourages diversity among the items selected in the batch.

6.3.1 The BatchGreedy Algorithm

Using this notation, the BatchGreedy policy will greedily select the i-th element in the j-th batch

$$v_{i,j} = \arg\max_{v\in\mathcal{V}} \Delta_f(v \mid \{v_{1,j}, \ldots, v_{i-1,j}\}, \mathbf{x}_\mathcal{A}),$$

where $\mathbf{x}_\mathcal{A}$ is the set of observations (labeled examples) from batches up to $j-1$. After a batch is completed, all labels are requested and added to the observations $\mathbf{x}_\mathcal{A}$. Pseudocode is presented in Algorithm 7.

If we set the batch size k to 1, BatchGreedy reverts back to a fully sequential, greedy active learning scheme. In particular, for the active learning example, this algorithm reduces to the GBS policy. According to our discussion in §4.2.2, we know that cost of GBS is upper-bounded by $O(\log|\mathcal{H}|)$ times that of the optimal sequential policy. As the first main theoretical contribution of this chapter, we generalize their results, which only hold for fully sequential policies, to the batch setting.

Algorithm 7: The BatchGreedy algorithm.

Input: Quota Q. Objective f and prior $P(\mathbf{y}_\mathcal{V})$;
begin
  $\mathbf{x}_\mathcal{A} \leftarrow \emptyset$;
  while $f(\mathbf{x}_\mathcal{A}) \geq Q$ **do**
    $\mathcal{B} \leftarrow \emptyset$;
    foreach $i \in \{1, \ldots, k\}$ **do**
      $v \leftarrow \arg\max_{v'} \Delta_f(v' \mid \mathcal{B}, \mathbf{x}_\mathcal{A})$;
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{v\}$;
    end
    Observe $\mathbf{x}_\mathcal{B}$ and set $\mathbf{x}_\mathcal{A} \leftarrow \mathbf{x}_\mathcal{A} \cup \mathbf{x}_\mathcal{B}$;
  end
end

6.3.2 Theoretical Analysis

We first show that BatchGreedy is near-optimal as compared to the optimal batch selection policy.

Theorem 6.8. *Let* $\text{cost}_{avg}(\pi^*_{batch,k})$ *be the expected cost and* $\text{cost}_{wc}(\pi^*_{batch,k})$ *be the worst-case cost of an arbitrary policy* $\pi^*_{batch,k}$ *selecting batches of size* k*. Further let* $\delta = \min_{\mathbf{x}_\mathcal{V} \in supp(\mathbb{P})} \mathbb{P}[\mathbf{x}_\mathcal{V}]$*. Then for the cost of the policy* $\pi^g_{batch,k}$ *implementing* BatchGreedy *it holds that*

$$\text{cost}_{avg}(\pi^g_{batch,k}) \leq \text{cost}_{avg}(\pi^*_{batch,k}) \left(\frac{e}{e-1}\right) \left(\ln Q + 1\right), \text{and}$$

$$\text{cost}_{wc}(\pi^g_{batch,k}) \leq \text{cost}_{wc}(\pi^*_{batch,k}) \left(\frac{e}{e-1}\right) \left(\ln \frac{Q}{\delta} + 1\right).$$

Note that the guarantee of Theorem 6.8 matches (up to a small constant factor) hardness results known for the fully sequential ($k = 1$) setting, which it generalizes, therefore BatchGreedy is near-optimal under computational constraints. Further note that for the active learning application, Theorem 6.8 guarantees that in the non-Bayesian setting (i.e., without any prior[3]), BatchGreedy requires at most a factor of $\mathcal{O}(\ln |\mathcal{H}|)$ more batches than the optimal batch-mode policy.

[3] In the special case of a uniform prior, $\delta = \frac{1}{|\mathcal{H}|}$. For general priors (with small δ), BatchGreedy can be proved to yield an $O(\log |H|)$ approximation, following analogously from Theorem 9.1 in Golovin and Krause [GK11b].

As the second main theoretical result of this chapter, we also prove that, perhaps surprisingly, under certain conditions BatchGreedy is not just competitive with an optimal policy that is restricted to selecting batches of examples: It is competitive with respect to an optimal fully sequential policy, which is *not* required to obey such a restriction.

Theorem 6.9. *Fix $\beta > 0$. Let $\text{cost}_{wc}(\pi^*_{seq})$ be the worst-case cost of an optimal* sequential *policy π^*_{seq}, constrained to picking a number of items which is a multiple of k. For the cost of the policy $\pi^g_{batch,k}$ implementing* BatchGreedy*, run until it achieves $f(\pi^g_{batch,k}) \geq Q - \beta$ it holds that*

$$\text{cost}_{wc}(\pi^g_{batch,k}) \leq \text{cost}_{wc}(\pi^*_{seq}) \cdot \textsc{gap}_k \cdot \frac{e}{e-1} \cdot \left(\ln \frac{Q}{\beta} + 1\right).$$

Moreover, it holds that $P\big(f(\mathcal{S}(\pi_G, \mathbf{x}_\mathcal{V})) \geq Q\big) \geq 1 - \beta$.

The proofs are given in §A.4. The key technical insight behind the proof relies on the adaptive gap proved in §6.2, which allows us to interpret the BatchGreedy policy as an approximate implementation of the fully sequential greedy policy. Note that Theorem 6.9, for technical reasons, has a slightly different flavor than Theorem 6.8: it compares the optimal policy π^* always achieving quota Q with one achieving the quota Q only with probability $1 - \beta$. By choosing $\beta < \min_{\mathbf{x}_\mathcal{V} \in \text{supp}(P)} P(\mathbf{x}_\mathcal{V})$, it can be guaranteed that in fact $f(\mathcal{S}(\pi^g_{\text{batch},k}, \mathbf{x}_\mathcal{V})) = Q$ for all $\mathbf{x}_\mathcal{V}$ with nonzero probability.

Remark 6.10. Under the assumption that the test outcomes are independent, we can plug in the upper bound on $\textsc{gap}_k$ of Theorem 6.4 and obtain

$$\text{cost}_{wc}(\pi^g_{\text{batch},k}) \leq \text{cost}_{wc}(\pi^*_{\text{seq}}) \quad \left(\frac{e}{e-1}\right)^2 \left(\ln \frac{Q}{\beta} + 1\right).$$

In this case, the bound on the worst-case cost of Theorem 6.9 is only a factor of $e/(e-1) \approx 1.58$ larger than that of Theorem 6.8, irrespective of the batch size k.

Considering sublinear cost within a batch. In many practical applications, the cost of acquiring a batch of labels in parallel can be significantly less than the cost of obtaining the same number of labels in sequential rounds. This could be due to shared cost when obtaining a set of labels, such as overhead time for start-up when running experiments, communication cost when conducting a survey, etc. There are fixed up-front costs for each round (e.g., time/energy spent to initialize a (set of) experiments), as well as variable costs per test (e.g., there is an incremental cost when we conduct one

more experiment on the current platform). In such scenarios, instead of considering each test to have unit cost (i.e., by assuming a modular cost function), it is natural to model the cost as a *submodular function* of a set of examples selected in the same batch. One cost model that has been studied under the batch-mode setting [YC13] is to use a sublinear function of the batch size, i.e., $C(\mathcal{A}) = a|\mathcal{A}|^p + b$, where b is the set-up cost, $|\mathcal{A}|$ is the number of queries, a is the coefficient, and $p \leq 1$. If we consider this cost model and make no assumptions on the prior distribution, then by Theorem 6.7 and Theorem 6.9 we get

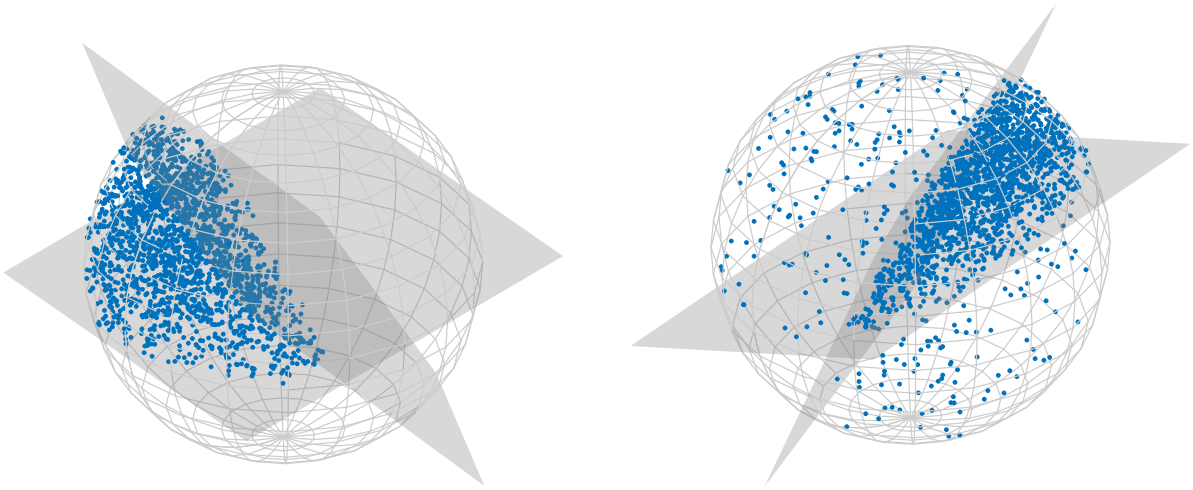
$$\mathrm{cost}_{wc}(\pi^g_{\mathrm{batch},k}) \leq \mathrm{cost}_{wc}(\pi^*_{\mathrm{seq}}) \left(\frac{e}{e-1}\right)\left(\ln\frac{Q}{\beta}+1\right)\cdot\frac{2\mathrm{GAP}_2}{\mathrm{GAP}_2-1}k^{\log \mathrm{GAP}_2}\cdot\frac{ak^p+b}{k}$$

Plugging in the upper bound on GAP_2 from Theorem 6.6, we obtain an upper bound on the cost of BatchGreedy under the sublinear cost model: $\mathrm{cost}_{wc}(\pi^g_{\mathrm{batch},k}) = \mathrm{cost}_{wc}(\pi^*_{\mathrm{seq}}) \cdot O\left(\left(\ln\frac{Q}{\beta}+1\right)k^{p-0.332}\right)$. We observe that the approximation factor is sublinear in k; for $p < 0.332$, BatchGreedy could incur lower cost than the optimal sequential policy when for large batch size k.

6.4 Efficient Implementation of BatchGreedy for BMAL

As is, BatchGreedy is not immediately practical for the two applications discussed in this chapter: Computing the marginal gains (6.3.1) requires computing expectations that may be intractable. In the influence maximization application, it is possible to perform Monte-Carlo sampling of the influence process to evaluate (6.3.1) up to arbitrarily small multiplicative error $(1+\varepsilon)$ [KKT03]. Furthermore, with a slight generalization of the arguments of Golovin and Krause [GK11b], using such an approximation of (6.3.1) increases the cost by at most the same factor $(1+\varepsilon)$.

To obtain a practical algorithm for batch mode active learning, further challenges arise: BatchGreedy requires that $\mathcal{H}$ is finite, and its running time depends polynomially on $|\mathcal{H}|$. Furthermore, it requires that observations are noise free. As a practical implementation, we focus on active learning of linear separators: let us denote the (κ-dimensional) feature vector of (unlabeled) example $v \in \mathcal{V}$ by $\mathbf{v} \in \mathbb{R}^\kappa$, then each hypothesis h corresponds to (homogeneous) linear separator $h(\mathbf{v}) = \mathrm{sign}(\mathbf{w}^T\mathbf{v})$. In our experiments, we use a Markov-Chain Monte Carlo sampler to generate samples from the posterior distribution over hypotheses $P(\mathbf{w} \mid \mathbf{x}_\mathcal{A})$. In particular, we build on the



(a) $\varepsilon = 0$ (b) $\varepsilon = 0.2$

Figure 6.3: Illustration of Algorithm 8 in 3-d space. (a) shows the sampling result in noise-free case (red arrows are constraints); (b) shows the sampling result when 20% of the observations are noisy: hypotheses that violate more constraints induce lower confidence.

hit-and-run sampler [Smi84; Lov98], which is known to lead to a provably efficient near-optimal estimation for the fully sequential active learning problem [GSSS11]. To handle noise, at each iteration, we generalize the hit-and-run sampler by sampling the entire version space, while varying the sample density according to a likelihood function. In the case of binary symmetric channel (i.e., labels are flipped with probability ε, c.f. Fig. 3.3), we sample hypotheses according to how many times they predict the wrong label. This way, our method can handle data that are not linearly separable. Algorithm 8 presents details of our sampler , and our final batch-mode active learning algorithm is formalized in Algorithm 9. The time complexity of random sampling (Algorithm 8) is $\mathcal{O}(sT)$, where s is the number of sampled hypotheses, and T is the number of mixing iterations. Once we discretized the hypothesis space with t samples, it takes $\mathcal{O}(kts)$ steps for Algorithm 9 to select a batch of k items (recall that t denotes the number of unlabeled examples). Hence the time complexity of Algorithm 9 selecting one batch is $\mathcal{O}(s(T + kt))$.

Furthermore, in both applications, we can use *lazy evaluations* to speed up the BatchGreedy algorithm (as used in Golovin and Krause [GK11b] for the fully sequential

Algorithm 8: (Noisy) hit-and-run hypothesis sampler for linear separators

Input: Labeled examples $\mathcal{B}$ and their labels $\mathbf{x}_\mathcal{B}$, the number of samples to be generated s, the number of mixing iterations T, the noise level $\varepsilon \in [0, 0.5)$;
Output: Hypotheses set $\hat{\mathcal{H}}$;
begin
  $\mathbf{w}_0 \leftarrow$ a random point on κ-dimensional unit sphere $\mathbb{S}^\kappa$;
  foreach $i \in \{1, \ldots, s\}$ **do**
    $\boldsymbol{\sigma} \leftarrow$ a random direction (unit vector) in $\mathbb{S}^\kappa$;
    Set $\mathcal{L} \leftarrow S^\kappa \cap \{\mathbf{w} \mid \mathbf{w} = \mathbf{w}_{i-1} + \boldsymbol{\sigma}\rho, \rho \in \mathbb{R}\}$, and select $\mathbf{w}_i$ from sector $\mathcal{L}$ with $\rho \sim p(\rho) \propto (\frac{\varepsilon}{1-\varepsilon})^l$, where $l = |\{v : v \in \mathcal{B}, (v, \text{sign}(\mathbf{w}_i^T \mathbf{v})) \notin \mathbf{x}_\mathcal{B}\}|$;
    Add every T-th sample $h_i(\mathbf{v}) = \text{sign}(\mathbf{w}_i^T \mathbf{v})$ to $\hat{\mathcal{H}}$;
  end
end

setting). Lazy evaluations utilize the fact that the marginal gains $\Delta_f(s \mid \mathcal{A}, \mathbf{y}_\mathcal{B})$ are monotonically decreasing in both $\mathcal{A}$ and $\mathbf{y}_\mathcal{B}$. This insight can be exploited by utilizing priority queues to accelerate selection of the next greedy choice.

6.5 Experimental Results

We empirically evaluate BatchGreedy on several data sets and both applications discussed at the beginning of this chapter. Our emphasis is on comparing BatchGreedy with baselines, as well as empirically quantifying the price of parallelism.

6.5.1 Multi-stage Influence Maximization in Social Networks

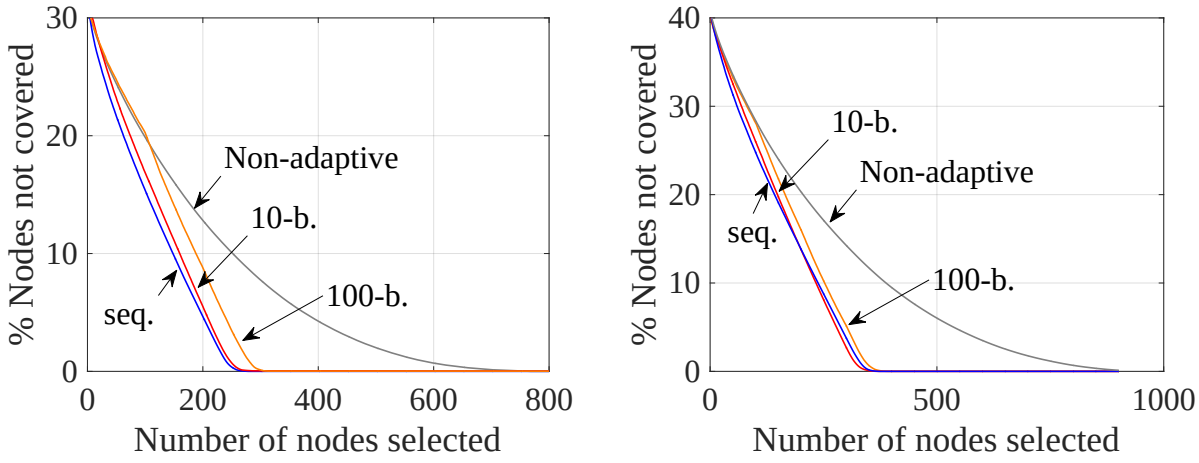
We first apply BatchGreedy to the multi-stage influence maximization problem. We use two data sets from the SNAP repository [LK14]: the *Epinions* social network (with 75879 nodes and 508837 directed edges, where members of the site can decide whether to "trust" each other) and the *Slashdot* social network (with 82168 nodes and 948464 directed edges, where users are allowed to tag each other as friends or foes). For each network, we take the subgraph induced by the top 1000 nodes with the largest outdegree. We use the independent cascade model [KKT03]. In our simulations, we

Algorithm 9: Approximate implementation of BatchGreedy for BMAL

Input: Hypotheses $\mathcal{H}$, batch size k, noise level ε;
begin
  Sample $\hat{\mathcal{H}} = \{h_1, \ldots, h_s\}$ from $\mathcal{H}$ using Algorithm 8;
  Define $\hat{P}(\mathcal{H}) = \frac{1}{s}\sum_{\ell=1}^{s} \delta_{h_\ell}$; set $\mathbf{x}_\mathcal{B} \leftarrow \emptyset$;
  while $(1-\varepsilon)$ *of all hypotheses in the support of* $\hat{P}$ *induce same labeling on the unlabeled pool* **do**
    $\mathcal{A} \leftarrow \emptyset$;
    foreach $i \in \{1, \ldots, k\}$ **do**
      $v \leftarrow \arg\min_{v'} \sum_{\ell=1}^{s} \left[\hat{P}(\mathcal{H}(\{(v, h_\ell(\mathbf{v})) : v \in \mathcal{A} \cup \{v'\}\}))\right]$;
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{v\}$;
    end
    Observe $\mathbf{x}_\mathcal{A}$ and set $\mathbf{x}_\mathcal{B} \leftarrow \mathbf{x}_\mathcal{B} \cup \mathbf{x}_\mathcal{A}$;
    Sample $\hat{\mathcal{H}} = \{h_1, \ldots, h_s\}$ (Algorithm 8) using $\mathbf{x}_\mathcal{B}$;
    Update approximation $\hat{P}(\mathcal{H}) = \frac{1}{s}\sum_{\ell=1}^{s} \delta_{h_\ell}$;
  end
end

assume that each person has a fixed probability of influencing its neighbors. We choose this probability t according to the edge density of the target network, in our case to 0.05 and 0.03, respectively.

We evaluate the performance of BatchGreedy while varying the size of batches picked at each stage. We repeat the experiments 100 times for all batch sizes (the non-adaptive method corresponds to infinite batch size). In each experiment, we initialize 100 random realizations of the target network based on the edge activation probability, and greedily select the best node in expectation. The results are summarized in Figure 6.4a and Figure 6.4b. We observe that for the *Epinions* network, the sequential greedy policy covers 99% of the target network by selecting 244 nodes, while the 10-batch greedy policy, 100-batch greedy policy and non-adaptive greedy policy cost 241, 284, and 584 nodes respectively, to achieve the same coverage. Similarly, for the *Slashdot* network those numbers are 330, 319, 343, 765. After acquiring the first batch of labels, BatchGreedy performs surprisingly well, even competitively with the fully sequential policy.



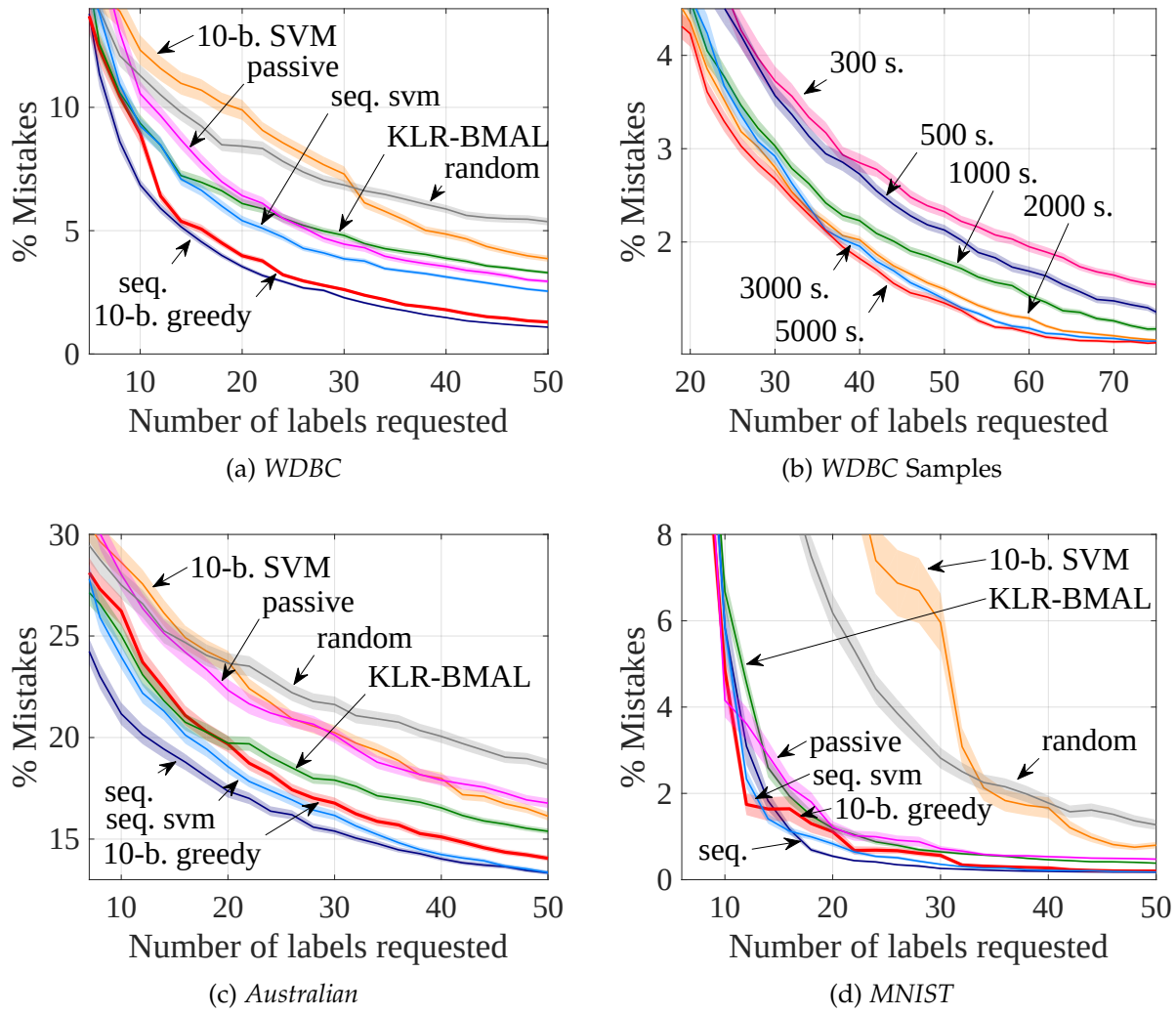
(a) *Epinions* (b) *Slashdot*

Figure 6.4: Results: Adaptive Influence Maximization

6.5.2 Batch-mode Active Learning of Linear Separators

One natural way to perform batch-mode active learning is to select batches comprising the k most uncertain examples. As one baseline, we use "batch mode margin-based active learning" (k-batch SVM) to greedily select batches of examples, as considered in Jain, Vijayanarasimhan, and Grauman [JVG10] for large-scale active learning. In this method, we randomly chose examples until there are two distinct labels, and we train SVM classifiers based on the labeled examples at the end of each batch. The next k unlabeled examples with the smallest distances from the decision boundary $\mathbf{w}^T\mathbf{v} + b = 0$ are selected for labeling. Another baseline approach we employ is the state of the art batch mode active learning algorithm (KLR-BMAL) of Hoi et al. [Hoi+06b] that selects batches of k examples that are informative w.r.t. the Fisher information matrix. To see how well the parallelization of the selection process approximates the sequential algorithm, we compare it with the fully sequential active learning algorithm, where only one example is selected and observed at each iteration, as well as a "passive learning" approach, where we make no observations during the learning process (corresponding to infinite batch size). We also compare BatchGreedy against the sequential algorithm with purely random selection.

For a fair comparison, we use SVM as the classifier for all competing algorithms, so the methods only differ by the set of examples chosen for labeling. We implement the KLR-BMAL algorithm using class membership probabilities inferred from the



(a) *WDBC* (b) *WDBC* Samples

(c) *Australian* (d) *MNIST*

Figure 6.5: Results: Batch-mode active learning

hypothesis sampler and set the smoothing parameter δ to be 0.1 [Hoi+06b]. As for our sampler, we set ε to be 0.1 (Generally ε can be chosen via cross-validation). We normalize the data so that each feature has mean 0 and standard deviation 0.5, and place independent normal priors on each dimension. The results of all the batch mode active learning experiments are obtained from 150 random starts.

We run our first set of experiments on two UCI datasets [Lic13], *WDBC* (569 instances, 32-dimensional) and *Australian* (690 instances, 16-dimensional), using a fixed number of 5000 sampled hypotheses in each random trial. Figure 6.5a and Figure 6.5c depict the 150-trial average percentage of mistakes made by each algorithm when predicting the labels of the corresponding data set, for a batch size of $k = 10$. Figure 6.5c shows an

improvement of BatchGreedy over both KLR-BMAL and the 10-batch SVM algorithm. On both datasets, surprisingly, BatchGreedy is competitive with the fully sequential greedy algorithm, with only minor differences.

We also evaluate BatchGreedy on the *MNIST* dataset. For each of the 14780 instances, we reduced the dimensionality down to 10 via PCA, and compare BatchGreedy with the sequential, KLR-BMAL, 10-bacth SVM, passive and random algorithms through 150 random trials. We observe that, even using 5000 sample hypotheses for each iteration, BatchGreedy is significantly faster than KLR-BMAL, as the cost of BatchGreedy grows linearly w.r.t. the number of hypotheses and number of examples, while KLR-BMAL costs quadratically w.r.t. the number of examples. In fact, for the same settings, it takes KLR-BMAL approx. 50 seconds to select one example, compared to approx. 10 seconds for BatchGreedy.

Furthermore, we study the impact of the discretization parameter M (i.e., the number of hypotheses used to sample the version space), varying it from 300 to 5000, and we plot the results for each setting in Figure 6.5b. For the *WDBC* data set, we can observe a statistically significant performance improvement across the 150 trials when increasing the number of sampled hypotheses used from 300 to 2000. Starting from 3000 samples, however, the advantage of introducing more samples begins to decrease dramatically. As there is a linear increase in running time as we employ more samples, we suggest to pick a moderate M to balance efficiency and accuracy.

6.6 Summary

We presented a general framework for batch mode active learning and stochastic optimization. We analyzed BatchGreedy, an intuitive adaptive greedy approach, and proved its competitiveness with the optimal batch-mode policy. For some problem instances (e.g., multi-stage influence maximization) we proved that perhaps surprisingly, the use of batches only incurs a bounded increase of cost as compared to allowing fully sequential selection. In addition to new theoretical results, we empirically demonstrate the effectiveness of BatchGreedy on two real-world applications: Batch mode active learning of linear separators (where BatchGreedy outperforms state of the art), and multi-stage influence maximization (where we observe a surprisingly small increase in cost compared to the fully sequential strategy). A natural question for future work is to

understand more generally for which problems the price of parallelism, i.e., the increase in cost by restricting to information-parallel decisions, is bounded. We believe that our results provide an important step in characterizing the (approximate) tractability of practical active learning and optimization problems.

7

Unknown parameters: Converting Offline to Online

A crucial assumption we have made in Part II of this dissertation is that the probabilistic model is fully specified, and thus we can score tests based on such information for greedy optimization. Unfortunately, in practice, the underlying distribution over hypotheses/test outcomes $\mathbb{P}[\mathbf{X}_{\mathcal{V}}]$ is often unknown and requires to be learned over time. For instance, in viral marketing/ influence maximization (Example 6.1), we may not know in advance the probability of a node successfully influencing its neighbors; in active learning (Example 6.2), we may not have access to prior distribution over the hypotheses. Since our adaptive greedy framework described in Algorithm 1 relies on the prior distribution over the test outcomes, it is unclear how one can adapt our algorithms to handle this challenging practical setting.

In this chapter, we propose an *online* adaptive information acquisition framework, which adapts the offline algorithms considered in this dissertation into the online setting. Concretely, we maintain a distribution over the *parameters* of $\mathbb{P}[\mathbf{X}_{\mathcal{V}}]$; for each *epoch*, we employ a *posterior sampling* approach, where we first draw samples of the parameters of the distribution according to their probabilities of being "true" conditioned on the current observation (i.e., in the sense that they reflect the true parameters), and then deploy the adaptive greedy framework on top of the sampled distribution.

We establish a rigorous bound on the *expected regret* (defined in terms of the value of information) of the online algorithm. As a concrete application, we demonstrate our online learning framework on the troubleshooting platform which was introduced in §4.6.4. Our experimental results imply that the online framework encourages active exploration, which, combined with the offline algorithms proposed in §4, leads to effective online learning of the optimal VoI.

Organization of this chapter. We describe the problem setup, and formally state the online optimization problem in §7.1. In §7.2, we propose OnlineVoI, our efficient online framework for adaptive information acquisition. In §7.3 we present our main theoretical result on the regret bound of OnlineVoI. We demonstrate the effectiveness of the framework in §7.4 and summarize the chapter in §7.5.

7.1 Problem Statement

We first provide two motivating applications for the online learning setting, and then formally state the key problem of this chapter.

7.1.1 Motivating Applications

Example 7.1 (Online interactive troubleshooting)**.** *Imagine that we want to build an intelligent online troubleshooting system as a replacement of human representatives at a customer service/ call center of a cellphone company. At an early stage, we do not have full knowledge on what symptoms correspond to what root-causes of the problems of a cellphone. Rather, we start with some prior estimation of the distribution over the (root-cause, symptom) pairs, and would like to refine the model online. Every time a customer calls in, the system can ask questions on the symptoms of her device and receive feedback. After acquiring enough information on the symptoms, the system suggests what is the root-cause and solution to the customer at the end of the conversation, and receive a reward. We call each conversion session an "epoch" in the online learning process. The solution can be further verified by a human agent, so that at the end of each epoch, the system obtains a sequence of (root-cause, symptom) pairs which it can use as new training data. The goal is to design an online sequential decision making strategy which attains high reward while minimizing the number of questions asked.*

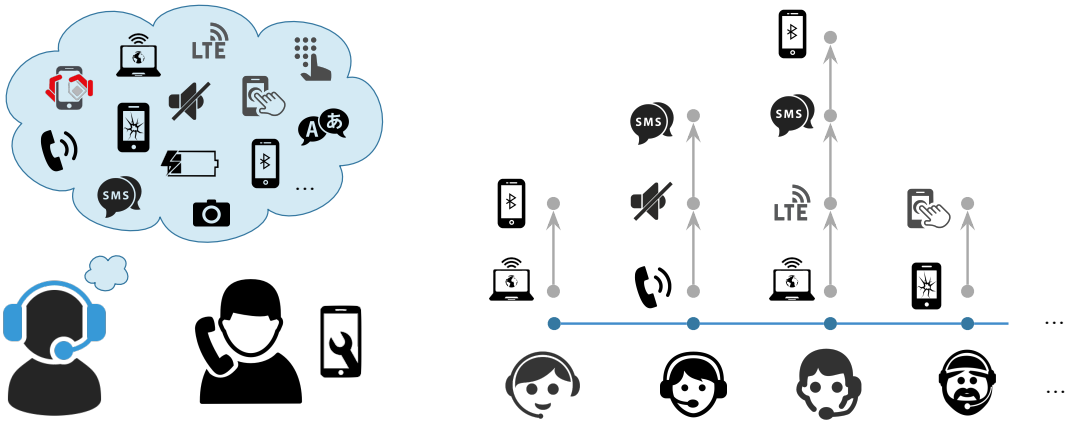


Figure 7.1: Converting offline to online.

We illustrate our problem setup in Fig. 7.1. Customers call in one after the other for troubleshooting. Within each epoch (i.e., for each client), we are essentially solving an (offline) adaptive information acquisition problem, which we have discussed extensively in §4. The reward that the system can get at the end of each epoch is exactly the decision-theoretic value of information (c.f. Eq. (2.2.3) and (4.1.1)). If we know the true distribution, we know from §4 that running DIRECT (§4.3) or ECED (§4.5) guarantees near-optimal cost for achieving the maximal reward, when comparing with the optimal algorithm under the actual distribution. Under the online learning setting, other than asking the most informative question for solving the current customer's problem (i.e., exploitation), we also need to engage the client with more explorative questions that may help with learning the probabilistic model (i.e., exploration).

Example 7.2 (Online Viral Marketing/ Information Propagation)**.** *Our second example concerns a viral marketing/ information propagation problem over a social network. The network is represented as a weighted directed graph, where the weight of each directed edge represents the edge "activation probability" (i.e., the probability that information can propagate through the edge). Different from Example 6.1, we assume that the edge activation probabilities are unknown, and need to be learned online. Now suppose that we want to promote* a series of *products to (a fraction of) the network. For each product, we choose to adaptively give a few samples for free to a subset of people, each based on the observed influence in the previous rounds. Our goal is to design on online adaptive strategy for choosing which node to target, so that we can influence a certain fraction of the network, while minimizing the budget spent on advertising.*

In this example, the reward is the total number (or fraction) of nodes influenced by

targeting a subset of nodes. Due to lack of knowledge of the diffusion model, we cannot directly compute the expected influence of a node in the same way we do for Example 6.1 in the offline setting. We have to give away more products to learn the weight of the graph. Analogously to Example 7.1, we need to trade-off exploration (i.e., giving away products to uncertain, possibly low-influence nodes to learn the weight of its incident edges) and exploitation (i.e., targeting highly influential nodes to maximally spread the information).

7.1.2 General Problem Statement

Following the notations defined in §2.1, we use $\mathcal{V} = \{1, \ldots, t\}$ to denote the set of tests, use $\mathbf{X}_{\mathcal{V}}$ to denote the random outcomes of the tests in $\mathcal{V}$, and use $\mathbf{x}_{\mathcal{V}}$ to denote a specific realization of $\mathbf{X}_{\mathcal{V}}$. Given a policy π, we use $\mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}})$ to denote the set of tests (and their outcomes) seen by π under realization $\mathbf{x}_{\mathcal{V}}$. In this chapter, we assume that tests have unit cost, i.e., $\forall v \in \mathcal{V}$ we have $c(v) = 1$, and we use $f(\mathbf{x}_{\mathcal{A}})$ to denote the *value* or the *reward* of the observation $\mathbf{x}_{\mathcal{A}}$.

Under the online setting, the prior distribution $P := \mathbb{P}[\mathbf{X}_{\mathcal{V}}]$ is unknown, and hence we have to select tests based on some estimation of P. Let $\tilde{P}_\ell$ denote the estimated distribution over $\mathbf{X}_{\mathcal{V}}$ at the start of the ℓ^{th} epoch, and π_ℓ denote the associated adaptive policy we are running. In principle, we want to design an adaptive policy for each epoch that is competitive with the *hindsight-optimal* policy that knows the true distribution. We define the *regret* over the ℓ^{th} epoch as

$$\tilde{\Delta}_\ell = F(\pi^*) - F(\pi_\ell),$$

where $F(\pi) \triangleq \mathbb{E}_{\mathbf{x}_{\mathcal{V}} \sim P}[f(\mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}}))] = \sum_{\mathbf{x}_{\mathcal{V}}} P(\mathbf{x}_{\mathcal{V}}) f(\mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}}))$ denotes the expected value of policy π under the *true* distribution P, and $\pi^* = \arg\max_\pi F(\pi^*)$ denotes the optimal policy which achieves the best value under the true distribution P.

Suppose we have a fixed budget τ for each epoch. We define the *accumulated regret* incurred by running $\{\pi_1, \ldots, \pi_k\}$ over k epochs as

$$\text{Regret}(k, \tau) = \sum_{\ell=1}^{k} \tilde{\Delta}_\ell. \tag{7.1.1}$$

We will assess the algorithm performance in terms of the expected accumulated regret.

7.2 Online Learning for Optimizing VoI

We employ a *posterior sampling* strategy [ORVR13], described as follows. Suppose that initially we have access to a prior over the model parameters. For example, in the troubleshooting application, we maintain Beta prior distributions $B(\boldsymbol{\alpha}, \boldsymbol{\beta})$ on the parameters $\boldsymbol{\varrho} := [\varrho_{ij}]_{t \times m}$ of the conditional probabilities, where recall $\varrho_{ij} = \mathbb{P}\left[X_i = 1 \mid \theta_j\right]$, and t (resp. m) represents the number of tests (resp. root-causes). That is, $\forall i \in [t], j \in [s]$, we have $\varrho_{ij} \sim B(\alpha_{ij}, \beta_{ij})$ where α_{ij}, β_{ij} depend on historical data. Similarly, for viral marketing, $\boldsymbol{\varrho}$ corresponds to the vector of edge activation weights, and we can assume a Beta prior on each element of $\boldsymbol{\varrho}$.

At the start of epoch ℓ, we sample $\boldsymbol{\varrho}^{(\ell)}$ from the posterior distribution over the parameters, conditioned on the observation history $\textsc{Obs}_{\ell-1}$, and obtain $\tilde{P}_\ell$. We then run the adaptive greedy policy π_ℓ (i.e., Algorithm 1) over epoch ℓ according to the sampled distribution $\tilde{P}_\ell$. When the epoch is over, we obtain observations $\textsc{Obs}_\ell$, based on which we update the posterior, and continue. We call this algorithm OnlineVoI, and provide the pseudocode in Algorithm 10.

Algorithm 10: The online framework for adaptive information acquisition

Input: Prior $\mathbb{P}[\boldsymbol{\varrho}]$ over the parameters $\boldsymbol{\varrho}$ of distribution $\mathbb{P}[\mathbf{X}_v]$; reward function f;
begin
  foreach $\ell = 1, 2, \ldots$ **do**
    Sample $\boldsymbol{\varrho}^{(\ell)} \sim \mathbb{P}[\boldsymbol{\varrho} \mid \textsc{Obs}_{\ell-1}]$ and obtain distribution $\tilde{P}_\ell$;
    Call the adaptive greedy framework with $(f, \tilde{P}_\ell)$ to run π_ℓ;
    Add the set of observations to $\textsc{Obs}_\ell$;
  end
end

Our online learning strategy can be interpreted as Thompson sampling across multiple epochs of interaction. Several recent empirical simulations [CL11; Gra+10; Sco10] and theoretical studies [AG12; BL13b; KKM12] have demonstrated the effectiveness of Thompson sampling in different settings. The classical usage of Thompson sampling [Tho33] suggests to choose an action according to its probability of being optimal, whereas, in our model, an "action" can be interpreted as the set of tests performed in one epoch.

7.3 Theoretical Analysis

To make our analysis concrete, in the following, we focus on the online variant of the decision-making problem discussed in §4. This problem setting captures the troubleshooting application (Example 7.1). For this specific problem, we instantiate the generic OnlineVoI algorithm as below.

7.3.1 OnlineVoI for Decision Making

Algorithm 11: Online adaptive information acquisition for decision making

1 **Input**: α_{ij}, β_{ij} parameters of Beta distributions, prior over root-causes $\mathbb{P}[\Theta]$, ;
begin
2 **foreach** $\ell = 1, 2, \ldots$ **do**
3 $\mathcal{A} \leftarrow \emptyset, \mathbf{x}_{\mathcal{A}} \leftarrow \emptyset, r \leftarrow$ NaN;
4 Draw $\boldsymbol{\varrho}^{(\ell)} = \{\varrho_{ij}^{(\ell)} \sim B(\alpha_{ij}^{(\ell)}, \beta_{ij}^{(\ell)})\}$, and obtain $\tilde{P}_\ell$;
5 Call the submodular surrogate-based greedy framework (Algorithm 3) to engage epoch ℓ with policy π_ℓ, and obtain a sequence of (test, outcome) pairs $\mathbf{x}_{\mathcal{A}}$;
6 Observe the root-cause for epoch ℓ, with index ϑ;
  foreach $(v, x_v) \in \mathbf{x}_{\mathcal{A}}$ **do**
7 Update $\alpha_{v\vartheta}^{(\ell)}, \beta_{v\vartheta}^{(\ell)}$ to get $\alpha_{v\vartheta}^{(\ell+1)}, \beta_{v\vartheta}^{(\ell+1)}$;
  end
end
end

For simplicity, we assume that the prior $\mathbb{P}[\Theta]$ over root-causes is given. In principle, we can drop this assumption, and instead assume a prior over the parameters of $\mathbb{P}[\Theta]$ so that we can sample from it, similarly as how we sample $\boldsymbol{\varrho} \sim B(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

7.3.2 Regret Bound for OnlineVoI

Suppose we have drawn $\boldsymbol{\varrho}^{(\ell)}$ to obtain distribution $\tilde{P}_\ell$ for epoch ℓ. To compute π_ℓ (i.e., to run Algorithm 3 at Line 5 of Algorithm 11), we first need to run the dynamic

hypothesis enumeration algorithm (c.f. §4.4) to generate samples $\tilde{\mathcal{H}}_\ell$ that covers $1-\eta$ fraction of the probability mass of the hypothesis space, and construct a DRD problem instance on top of $\tilde{\mathcal{H}}_\ell$. Denote the optimal policy with respect to distribution $\tilde{P}_\ell$ by $\pi^*_{\tilde{P}_\ell}$. By the definition of the DRD/ NVOI-NMU problem (Problem 4.1.2), we know that the value of all *feasible* policies for Problem 4.1.2 is at most ε away from the optimal value.

W.l.o.g, we assume in this section that $\varepsilon = 0$, and that decision regions in the DRD problem are *disjoint*. Let $\tilde{p}_{\min,\ell} = \min_{h\in\tilde{\mathcal{H}}_\ell} \frac{\tilde{P}(h)}{1-\eta}$. Then, Theorem 4.15 implies that with probability at least $1-\eta$, running π_ℓ (i.e., Algorithm 3) at epoch ℓ achieves the same value of information with $\pi^*_{\tilde{P}_\ell}$, with at most $(2\ln(1/\tilde{p}_{\min,\ell})+1)$ times of the minimal (worst-case) cost.

Now suppose we run Algorithm 11 for k epochs. Define $c^* \triangleq \max_{\ell,\mathbf{x}_\mathcal{V}} |\mathcal{S}(\pi^*_{\tilde{P}_\ell}, \mathbf{x}_\mathcal{V})|$ to be the worst-case cost of the optimal policies over any of the k epochs, and $\delta \triangleq \min_\ell \tilde{p}_{\min,\ell}$ to be the minimal probability of any sampled hypothesis. We establish the following bound on the expected accumulated regret of running Algorithm 11:

Theorem 7.3. *Fix $\eta \in (0,1)$. Let $\tau = (2\ln(1/\delta)+1)\,c^*$ be the length of each epoch, where δ and c^* are defined as above. Then, running Algorithm 11 over k epochs achieves expected accumulated regret*

$$\mathbb{E}[\text{Regret}(k,\tau)] = O\left(\tau S\sqrt{tk\tau\log(Stk\tau)} + k\eta\right), \tag{7.3.1}$$

where S is the total number of possible realizations of τ tests, and t is the number of tests.

We defer the proof of Theorem 7.3 to §A.5. The main idea behind the proof is to view the DRD/ NVOI-NMU problem as optimizing a (finite horizon) Partially Observable Markov Decision Process (POMDP) over repeated episodes of a fixed horizon τ. The parameter S in the regret bound corresponds to the number of (reachable) belief states of the POMDP. Each belief state in an episode represents a (sub-) set of selected tests and observed outcome of each test. Once we have established this connection, we can interpret the online learning problem as a reinforcement learning problem via posterior sampling, in a similar way to Osband, Russo, and Van Roy [ORVR13].

Remark 7.4. A conservative bound on S in $2^{\binom{t}{\tau}}$, which is doubly-exponential in the time horizon of each episode τ. However, in practice, the number of reachable belief states is limited by the structure of the problem (e.g., configuration of the CPTs), and hence

could be far smaller. In any case, (by applying Markov's inequality) Eq. (7.3.1) implies that

$$\frac{\text{Regret}(k,\tau)}{k\tau} \to \frac{\eta}{\tau}.$$

In words, the regret of Algorithm 11 *in the limit* (as $k \to \infty$) is bounded by η/τ.

Remark 7.5. If $\tilde{p}_{\min,\ell}$ is too small (and hence δ is too small), we can use a similar trick proposed by Kosaraju, Przytycka, and Borgstrom [KPB99b] (as discussed in §4.2.2 for improving the bound for GBS) to run the greedy algorithm on a modified prior $\tilde{P}'$ over $\tilde{\mathcal{H}}_\ell$, i.e., by setting $\tilde{P}'(h) \propto \max\left\{\frac{\tilde{P}(h)}{1-\eta}, \frac{1}{|\tilde{\mathcal{H}}_\ell|^2}\right\}$. In these cases, a lower bound on $\tilde{p}_{\min,\ell}$ is $\max\left\{\min_{h\in\tilde{\mathcal{H}}_\ell} \frac{\tilde{P}(h)}{1-\eta}, \frac{1}{|\tilde{\mathcal{H}}_\ell|^2}\right\}$, which consequently puts a lower bound on δ.

Remark 7.6. As mentioned in §2.4, one can view the general class of adaptive information acquisition problems as probabilistic planning problems. Hence, similar analysis also applies to other problems (e.g., the online viral marketing problem of Example 7.2) which involves solving adaptive submodular minimum cost coverage problem in each epoch.

7.4 Experimental Results

Dataset and experimental setup. To keep this chapter practically grounded, we evaluate our online framework on an *online* interactive troubleshooting application. We conduct experiments on the same experimental platform as described in §4.6.4, where we have evaluated the offline algorithms. Recall that there are a total number of 1100 root-causes and 950 binary tests in our dataset. To create an initial distribution over the conditional probabilities $\boldsymbol{\varrho}$, we set Beta priors $B(\alpha_{ij}, \beta_{ij})$ on $\boldsymbol{\varrho}_{ij}$. The ratio α_{ij}/β_{ij} is set to be roughly proportional to the ratio between the number of $(x_i = 1, \theta_j)$ pairs and the number of $(x_i = 0, \theta_j)$ pairs in the training set. For robustness, we further inject noise into these estimates by flipping the zero entries of α_{ij}'s and β_{ij}'s with some small probability. In our experiments, we assume that the prior distribution over root-causes is given, which we assume to be uniform.

We evaluate Algorithm 11 over 100,000 simulated test scenarios. Each test scenario corresponds to a root-cause θ and an underlying hypothesis h. As with §4.6.4, our goal is to identify the root-cause, meaning that each root-cause θ corresponds a decision y. We accept a "give-up" decision region, and therefore the resulting decision regions are

disjoint (intuitively, hypotheses in the intersection between multiple decision regions are assigned to the "give-up" decision). We adopt the same utility functions $u(\theta, y)$ as used for the offline setting to construct regions for solving the DRD problems, i.e.,

$$u(\theta, y) = \begin{cases} 0, & \text{if } y \text{ is "give-up";} \\ 1, & \text{if } y \text{ matches } \theta, \text{ i.e., } y \text{ is the correct decision} \\ -19, & \text{if } y \in \text{supp}(\Theta) \wedge y \neq \theta, \text{ i.e., } y \text{ corresponds to a wrong root-cause} \end{cases}$$

We compare with the same set of baseline criteria as used in §4.6.4, namely the Information Gain (IG) criterion, and the Uncertainty Sampling (US criterion.

Results. Instead of keeping a fixed budget for each test scenario (i.e., epoch), we set a maximal budget for each test scenario, and keep running the policy until it identifies a decision region or exceeds the budget. At the end of epoch ℓ, we report the average label complexity (i.e., the accumulated means of the number of questions asked over ℓ epochs) and the average utility (i.e., the accumulated means of the utility over ℓ epochs).

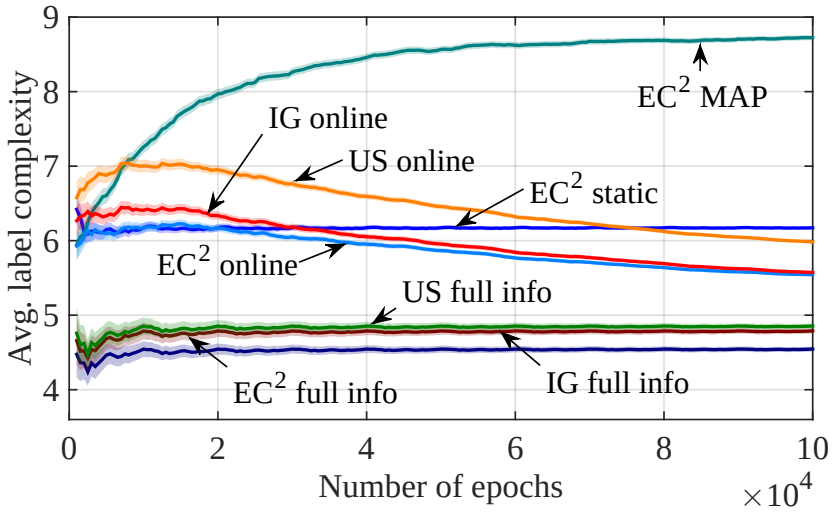


Figure 7.2: Results: Average cost of online interactive troubleshooting.

Fig. 7.2 and Fig. 7.3 demonstrate the behavior of different selection criteria. In "EC^2 MAP", we run Algorithm 11, but set ϱ to be their MAP estimators at Line 4. We can see its cost is much higher as it does not encourage exploration as much as the other

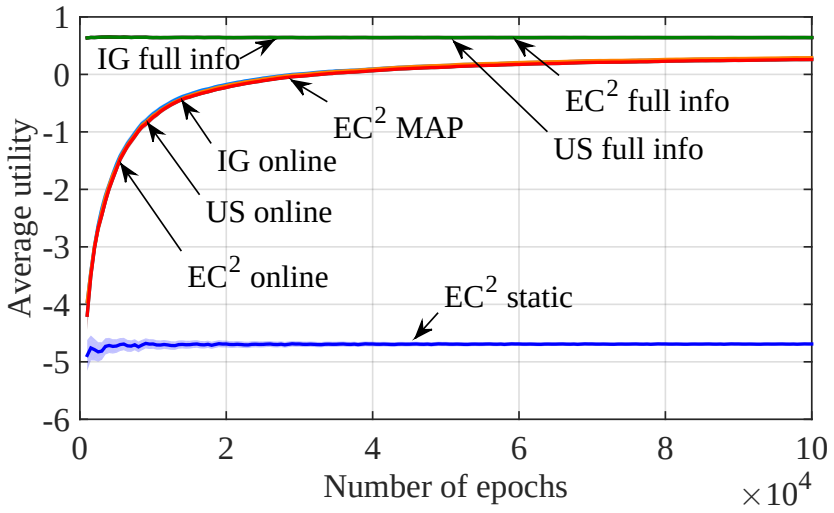


Figure 7.3: Results: Average utility of online interactive troubleshooting.

online methods. In "EC^2 static", we run EC^2 only based on the initial sample of CPT without any updates (i.e., skipping Line 7 of Algorithm 11). In the "full info" versions of the algorithms, the ground truth ϱ is used, while in the "online" versions, we use Algorithm 11 to update ϱ. Two notable observations are in order: first, with our online framework, the average utility (a.k.a reward) of all algorithms approach the optimal utility over time. Second, the EC^2 variants consistently outperform the alternatives in terms of query complexity, which is consistent with the results in the offline setting (i.e., assuming we have full knowledge of the model parameters).

7.5 Summary

We looked into a practical setting for the optimal VoI problem, where the probabilistic model and prior distribution over hypotheses are unknown. To address this issue, we proposed an efficient, principled online learning framework, and proved that it attains low accumulated regret in the long run. We demonstrated promising empirical performance of our framework on a real-world troubleshooting platform. Our results suggest that the "posterior sampling"-based online learning framework can properly trade-off exploration and exploitation, and could be valuable for building real-world online adaptive information acquisition systems.

Part IV

Conclusion and Outlook

8

Conclusion and Outlook

In this dissertation, we have investigated the fundamental problem of adaptive information acquisition and decision making under uncertainty. We posed the following question:

> *How can we efficiently, adaptively acquire information under uncertain, noisy environment, so that we can attain the maximal value of information with the minimal cost?*

Previous theoretical results for adaptive information gathering are mostly limited to strong modeling assumptions that are not tailored to real-world applications, while most practical algorithms rely on greedy heuristics that have no performance guarantees. To ease the tension between theory and practice, we have developed principled and practical algorithms for a class of adaptive information acquisition problems. In particular, we have considered new applications (§4, §5), extensions (§6, §7), and alternatives (§3, §4) of the *adaptive submodular optimization* framework [GK11b], to handle more complex constraints in real-world problems.

While these results are encouraging, they also raise open questions. In this chapter, we highlight the key contributions presented in this dissertation and outline a few promising directions for future work.

8.1 Summary

In a nutshell, this dissertation aims to develop new theoretical insights for complex adaptive information acquisition problems, and implement these ideas on challenging real-world applications. We have looked into several theoretical variants of the problem, where it is challenging to characterize the value of information due to complex constraints and modeling assumptions, such as indirect information (§4), noisy feedback (§3,§4), delayed feedback in parallel systems (§6), and incomplete knowledge about the model (§7). We briefly elaborate on these points and summarize our key contributions below.

8.1.1 Adaptive Information Acquisition: A Theoretical Perspective

Submodular surrogates. The key analytical tool that motivates our research is (adaptive) submodularity. If we assume noise-free observations, many adaptive optimization problems enjoy such structural property, which allows for efficient and provably near-optimal solutions – this is particularly noteworthy since finding the optimal solution is provably hard. When a given objective function is not submodular (e.g., the decision-making problem we considered in §4), unfortunately, the greedy solution might provide arbitrarily poor solutions.

Inspired by the success of convex optimization in the statistical learning literature, we proposed adaptive submodular surrogates and demonstrated how to convert a non-submodular problem into submodular one in §4. Our key idea is to design a submodular surrogate function that properly captures the structure of the problem while being aligned with the target objective to be optimized. In this way, optimizing the surrogate has the same effect with optimizing the original objective, and a greedy algorithm provides a near-optimal solution.

Beyond adaptive submodularity. As a featuring component of most practical information acquisition problems, uncertainty is often an inseparable part of our models and observations. Understanding the dynamics of uncertainty plays a decisive role in the analysis of complex adaptive systems. In the presence of noisy observations, it is not clear how to relate an (adaptive submodular) surrogate to the original objective function (e.g., the reduction in entropy (§3), or the error probability (§4)), as they often do not

share the same optimality condition. Therefore, the adaptive submodular optimization framework is no longer (directly) applicable.

Instead of seeking submodular surrogates for solving those noisy problems, our approach combines theoretical insights behind adaptive submodular optimization and information theory (e.g., by using information-theoretic objectives (§3) and auxiliary functions (§4)) to formulate and reason about uncertainty. Even though our surrogate functions are no longer adaptive submodular, they satisfy a similar "approximately" diminishing returns condition, which allows us to derive approximation guarantees for efficient greedy algorithms. Our theoretical insight allows us to develop general, robust and practical algorithms that work well when the observations are noisy.

Batch-mode information acquisition and the price of parallelism. Besides the sequential setting, we also studied the more general "partially closed-loop" setting, where batches of tests are performed in parallel. The main theoretical questions addressed in §6 are how the best batch-mode policy compares to the best fully sequential policy, and (consequently) how to design efficient algorithms that perform provably well.

A natural way to quantify the "price of parallelism" is by the adaptivity gap. It is known that when test outcomes are independent, the adaptivity gap is bounded irrespective of the cardinality constraint. We looked into the more general setting and proved rigorous upper bounds for the case where tests outcomes can be dependent. We proved that a simple greedy policy is competitive with the optimal batch-mode policy. Furthermore, based on our results on the adaptivity gap, we provided trade-offs between the cost of our algorithm and the degree of parallelism. Our results imply that under certain assumptions on the underlying distribution and cost model, the greedy policy can be competitive with the optimal fully sequential policy.

Dealing with unknown parameters. We further considered an online extension of the adaptive greedy optimization framework (§7), where we assumed an unknown/incomplete the prior on test outcomes. We proposed to convert the offline adaptive algorithms to the online setting via posterior sampling, and proved an upper bound on the regret of the resulting online algorithms. The analysis applies as long as (1) the "reward" of each epoch can be measured by an adaptive monotone submodular function, and (2) in each session we are solving an adaptive submodular minimum cost coverage problem.

8.1.2 Applications Relevant to this Dissertation

From a practical standpoint, we have developed several real-world applications that fit well under our theoretical umbrella. In this section, we highlight a few:

Active touch-based localization on a robotic platform. One of the applications we have investigated is touch-based localization, which was developed to showcase our submodular surrogate-based optimization framework in §4. We have collaborated with researchers from the Robotics Institute at Carnegie Mellon University and implemented the HEC and DIRECT algorithms on a real robotic platform. We demonstrated both our algorithms on a robot manipulation task, where a robotic arm was programmed to localize the open button of a microwave oven. We vertically compared DIRECT against HEC, and showed that DIRECT achieved the state-of-the-art performance, in both computational complexity and query complexity.

Active object detection for biodiversity monitoring. In §5, we investigated *active object detection*, where an active learner needs to interact with domain experts till it identifies all object instances. We have collaborated with ecologists from the applied ecology department from ETH Zurich and the Liverpool John Moores University, who launched conservation drones to take high-quality photographs of wildlife habitats, in order to obtain accurate and timely data on the wildlife distribution in the surveyed area. To avoid the costly process of having a human expert going through all the images, we have built an active detection system based on ACTDET. Although the communication from human expert to the learner is a simple binary feedback, our results imply that by utilizing ACTDET, one can already benefit from including human-in-the-loop to substantially improve the state-of-the-art performance for object detection.

Online interactive troubleshooting. We have worked on an interactive troubleshooting application, based on real-world data collected from customer service center agents, in collaboration with researchers from Xerox Research Center Europe. We implemented the hypothesis enumeration algorithm from §4 for practical concerns and integrated it with the posterior sampling framework from §7. We demonstrated that the submodular surrogate-based offline adaptive algorithm, integrated into the online framework achieves low regret.

8.2 Open Problems and Future Work

We now discuss some open problems and propose a few directions for future work.

8.2.1 Dealing with More Complex Constraints

Adaptive information acquisition with non-modular cost In this dissertation, we mostly consider the cost function to be additive/ modular, and most of our analysis and algorithms are designed specific to the modular cost setting. In §6, we have brought up a batch-mode variant where cost is a *submodular function* of a batch of tests, e.g., due to shared cost and shared resources. Although we have discussed the performance of BatchGreedy under a specific submodular cost function, we did not design the greedy framework in the first place to handle the submodular cost. Furthermore, there are many other practical cost functions that are *supermodular*. Following this thread, an interesting question would be, can we design algorithms that are more tailored to non-modular cost functions?

Adaptive information acquisition with estimated prior In §7, we studied the online setting and provided some preliminary results on the performance of a simple posterior-based algorithm. Note that in our current regret bound, the horizon of an epoch is a function of the worst-case cost of the optimal policy that solves the Optimal VoI problem in each epoch – this could potentially be huge, in which cases our bound becomes trivial. To make the bound more realistic, it makes sense to compare with the optimal policy that only solves the Optimal VoI *with high probability*, similarly with what we have discussed in §4.5. One interesting future work along this line is to understand how the noise-resilient algorithms, such as ECED perform with estimated prior?

Bayesian active learning with delayed feedback A more general problem that encloses the batch-mode setting is the setting with delayed feedback. For example, in medical diagnosis, a doctor may perform some medical tests on a patient, but the results may only be revealed minutes, hours or even days later. As a second example, consider an active learning application where multiple labeling experts are available, but each expert needs some processing time to return a label. In both scenarios, we

aim to find the optimal testing policy (e.g., which tests to evaluate next, or which data instance to label and which expert to ask), to learn the underlying hypothesis (e.g., the root-cause of the patient's symptoms for medical diagnosis, or the optimal classifier for active learning) with the minimal cost.

Note in standard sequential active learning, the set of observations received at time step ℓ is a singleton. In the delayed model, however, the feedback that concerns the test at time ℓ is received at the end of the period $\ell + \tau$, i.e., it is delayed by τ time steps. Note that $\tau \equiv 0$ corresponds to the non-delayed case.

8.2.2 Beyond the Adaptive Information Acquisition Framework

We now briefly discuss more general adaptive systems that require intelligent interaction with the environment/human, which go beyond the adaptive information acquisition problem formalized in this dissertation. We call the general class of problems *interactive machine learning*.

Dealing with complex interactions in interactive machine learning. In this dissertation, we considered a basic interaction model for adaptive information acquisition, where the adaptive system/ learner proposed a test/ query, observe its outcome, and adapt to the observation. Interestingly, in real-world, for the general problem of *interactive machine learning* (IML), we could have much richer classes of interactions. To maximally utilize the bandwidth of the communication channel, it is important to make use of not only the *corrective feedback* (e.g., a "yes" or "no" answer), but also the *explanations* that may guide the algorithms for more efficient learning.

The first question comes along this line is, what forms might a query take? It is natural to think about active learning algorithms that can formulate queries in the language of rules, advice or explanations, in addition to simple labels for data points. There has been considerable interest in recent years in incorporating human domain knowledge into active learning algorithms. A follow-up question is, under what circumstances do we still have tractable near-optimal solutions?

Active learning of structured hypotheses from explanatory feedback. Most real-world applications are structured, in the sense that they are composed of multiple correlated random variables. For example, in computer vision, we might want to

predict the semantic category of each pixel; in natural language processing, we might be interested in parsing sentences syntactically. In the supervised learning setting, we assume that one has full access to the training set, i.e., all input data and their fully labeled structured output. The learning problem can then be formalized as find the maximal likelihood estimators for the model parameters, e.g., the weights of a log-linear model [Joa+09]. In the active learning setting, however, we are only given a subset of training examples, with (possibly partial) labels in the structured output space. Due to its inherent computational difficulties (as we have to deal with exponentially sized output spaces), active learning has been much less understood for structured prediction tasks.

Explanatory feedback often carries implicit structural information about the output labels; hence incorporating the explanatory feedback may dramatically reduce the uncertainty of the output space, making it feasible to analyze active learning for such complex hypotheses. As an example, consider a holistic scene understanding task in computer vision, where a learner queries the label of an image segment. Rather than return the partial label "*tree*", the domain expert may further provide support sentence s: "*because it appears above the ground and below the sky*." A fundamental challenge of including s is that the complexity of the problem increases by orders of magnitude. Therefore, we need to develop new algorithms for exploiting such structural information of explanatory feedback.

Query elicitation: interleaving active learning with machine teaching. In interactive, interpretable machine learning systems, learning and teaching are reciprocal and tightly coupled. On the one hand, the system is required to intelligently issue queries that are *informative* for learning complex hypotheses; on the other hand, the learner ought to be clear about what it means by the proposed queries, by providing additional explanations that ensure *interpretability*. This motivates us to weave together techniques developed for both active learning (i.e., in proposing the query) and machine teaching [Zhu15] (i.e., in generating the explanation of a query – in other words, the learner also needs to "*teach*" the domain expert why it issued the query). An interesting future work is to develop a macro framework that jointly optimizes the label complexity and the model interpretability. It also provides the opportunity to explore structured machine teaching settings that unify prior work on structured prediction and machine teaching, which has thus far focused on extremely simple concept classes.

Part V

Appendice

A

Proofs

A.1 Proofs from Chapter 3

A.1.1 Proof of Lemma 3.7: Relating $\mathbb{I}(\pi; Y)$ to $p_{v,\max}$

Lemma A.1 (Long version of Lemma 3.7)**.** *Consider random variables U and V where U takes values over the set $[t]$ with distributions p and V takes values over $[t']$ with distribution p'. We think of p (resp. p') as a row vector with size t (resp. t') which sums up to one. Furthermore, assume that $p' = pQ$ where $Q = [q_{i,j}]_{t\times t'}$ is a stochastic matrix and $q_{i,j} = \mathbb{P}[V = j \mid U = i]$, for $i \in [t]$ and $j \in [t']$. Let $Q_1, Q_2, \cdots, Q_t$ denote the rows of Q and define the minimum distance of Q as*

$$S = \Big(\min_{i,j\in[t]:i\neq j} |Q_i - Q_j|_{\mathrm{TV}}\Big)^2.$$

Also, define $p_{\max} = \max_{i\in[t]} p_i$ and $u_{\max} = \max_{i\in[t]} p_i(1-p_i)$. Then, we have

1. $\mathbb{I}(U;V) \geq S u_{\max}$.
2. $\mathbb{I}(U;V) \geq \frac{1}{2} S(1 - p_{\max})$.
3. *If $\mathbb{I}(U;V) \leq \theta S$, then we have $p_{\max} \geq 1 - 2\theta$. Also, if $\theta \leq \frac{1}{4}$, then $p_{\max} \geq \frac{1+\sqrt{1-4\theta}}{2}$.*

Proof. Let $p = [p_i]$ and $p' = [p'_i]$. Using the introduced notation, we have $\mathbb{P}[U = i, V = j] = p_i q_{i,j}$. We thus can write

$$\begin{aligned}
\mathbb{I}(U;V) &= \sum_{i=1}^{t}\sum_{j=1}^{t'} p_i q_{i,j} \log \frac{p_i q_{i,j}}{p_i p'_j} \\
&= \sum_{i=1}^{t} p_i \sum_{j=1}^{t'} q_{i,j} \log \frac{q_{i,j}}{p'_j} \\
&= \sum_{i=1}^{t} p_i D_{\mathrm{KL}}(Q_i || p') \\
&\geq 2\sum_{i=1}^{t} p_i |Q_i - p'|^2_{\mathrm{TV}}, \qquad \text{(A.1.1)}
\end{aligned}$$

where the last step is due to Pinsker's inequality [CK11, p 44].

For any $j \in [t]$ we can write

$$\begin{aligned}
\sum_{i=1}^{t} p_i |Q_i - p'|_{\mathrm{TV}}^2 &= p_j |Q_j - p'|_{\mathrm{TV}}^2 + \sum_{i:i\neq j} p_i |Q_i - p'|_{\mathrm{TV}}^2 \\
&\geq (1-p_j) p_j |Q_j - p'|_{\mathrm{TV}}^2 + p_j \sum_{i:i\neq j} p_i |Q_i - p'|_{\mathrm{TV}}^2 \\
&= \sum_{i:i\neq j} p_j p_i |Q_j - p'|_{\mathrm{TV}}^2 + \sum_{i:i\neq j} p_j p_i |Q_i - p'|_{\mathrm{TV}}^2 \\
&= \sum_{i:i\neq j} p_j p_i (|Q_j - p'|_{\mathrm{TV}}^2 + |Q_i - p'|_{\mathrm{TV}}^2) \\
&\geq \sum_{i:i\neq j} p_j p_i \frac{(|Q_j - p'|_{\mathrm{TV}} + |Q_i - p'|_{\mathrm{TV}})^2}{2}
\end{aligned}$$

By applying the triangular inequality for total variation distances, we obtain

$$\begin{aligned}
\sum_{i=1}^{t} p_i |Q_i - p'|_{\mathrm{TV}}^2 &\overset{(a)}{\geq} \sum_{i:i\neq j} p_j p_i \frac{|Q_j - Q_j|_{\mathrm{TV}}^2}{2} \\
&\geq \sum_{i:i\neq j} p_j p_i \frac{S}{2} \\
&= p_j (1-p_j) \frac{S}{2}. \qquad \text{(A.1.2)}
\end{aligned}$$

The proof of part (1) is then complete by combining (A.1.1) and (A.1.2).

For the second part of the lemma, assume w.l.o.g that $t = 2r+1$ and $p_1 \geq p_2 \geq \cdots \geq p_t$ (if t is even we can always let $t \leftarrow t+1$ and let $p_t = 0$).

We can then write

$$\begin{aligned}
\sum_{i=1}^{t} p_i |Q_i - p'|_{\mathrm{TV}}^2 &\geq \sum_{j=1}^{r} \left\{ p_{2j-1} |Q_{2_{j-1}} - p'|_{\mathrm{TV}}^2 + p_{2j} |Q_{2j} - p'|_{\mathrm{TV}}^2 \right\} \\
&\geq \sum_{j=1}^{r} p_{2j} \left(|Q_{2_{j-1}} - p'|_{\mathrm{TV}}^2 + |Q_{2j} - p'|_{\mathrm{TV}}^2 \right) \\
&\geq \sum_{j=1}^{r} p_{2j} \frac{\left(|Q_{2_{j-1}} - p'|_{\mathrm{TV}} + |Q_{2j} - p'|_{\mathrm{TV}} \right)^2}{2}
\end{aligned}$$

By applying the triangular inequality for total variation distances, we obtain

$$\sum_{i=1}^{t} p_i |Q_i - p'|_{\mathrm{TV}}^2 \geq \sum_{j=1}^{r} p_{2j} \frac{|Q_{2_{j-1}} - Q_{2j}|_{\mathrm{TV}}^2}{2} \geq \frac{S}{2} \sum_{j=1}^{r} p_{2j} \overset{(a)}{\geq} \frac{S}{4}(1-p_1).$$

Here, step (a) follows from the fact that when p_i's are decreasing, and we thus have $\sum_{j=1}^{s} p_{2j} \geq \frac{1-p_1}{2}$. Part 2 is now proven by using the above derivation and (A.1.1).

Part 3 simply follows from the fact that in the assumption $\mathbb{I}(U;V) \leq \frac{\theta S}{4}$ holds, then by part 2 we have $p_{\max} \geq 1 - 2\theta$. Also, if $\theta \leq 1/4$, then $p_{\max} \geq 1/2$ and from part 1 we get $p_{\max}(1 - p_{\max}) \leq \theta$ and putting the two together we get the result. □

A.1.2 Proof of Lemma 3.8: Lower Bound the Probability of the Event Λ

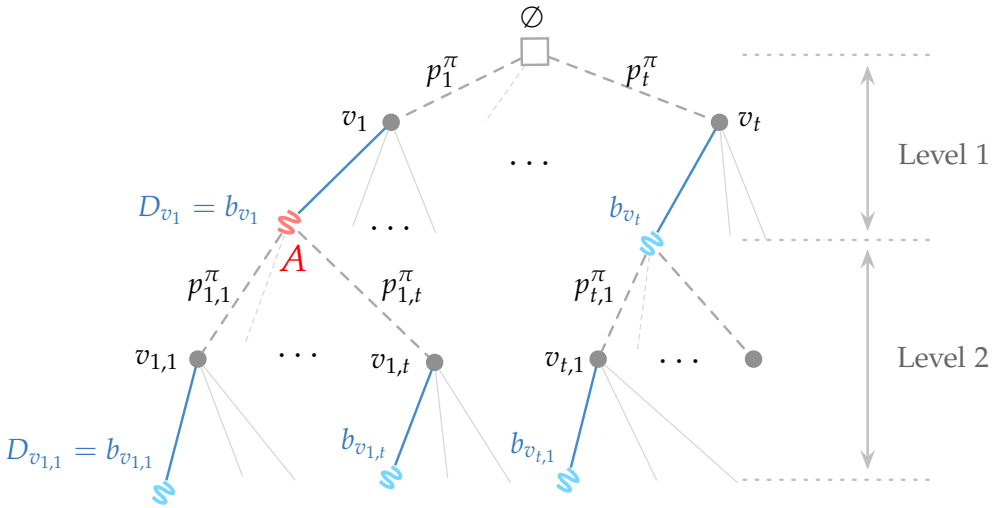


Figure A.1: A two-stage decision tree representation for (stochastic) policy $\pi_{(2)}$.

Proof of Lemma 3.8. Let us illustrate the idea by first assuming that π has length two (see Figure A.1). Afterwards we prove the statement for π with an arbitrary length k. Recall that we consider randomized policies too. In the very beginning, when no observations have been made, π can choose any of the possible t tests $v_1, \cdots, v_t \in [t]$. We thus assume that π chooses v_i with probability p_i^π. Furthermore, the choice of v_i clearly does not reveal any information about Y (because we only talk about the choice and hence no observations have been done so far). Let us define

$$\Lambda_r = \{(D_{v_1} = b_{v_1}) \wedge (D_{v_2} = b_{v_2}) \wedge \cdots \wedge (D_{v_r} = b_{v_r})\},$$

to be the event that the deterministic part of the first r tests that π has picked all have the most-likely outcome. Once π chooses its first test (let's say v_1), the event Λ_1 takes

place only if the output of D_{v_1} is precisely equal to b_{v_1}. Hence, in Figure A.1, to find $\mathbb{P}[\Lambda_1]$ we need to add up the probabilities of the blue paths up to level 1. As a result,

$$\begin{aligned}\mathbb{P}[\Lambda_1] &= \sum_{i=1}^{t} p_i^{\pi} \mathbb{P}[D_{v_i} = b_{v_i}] \\ &\geq \sum_{i=1}^{t} p_i^{\pi} \beta = \beta.\end{aligned}$$

Now, let us see what happens when π selects its second test. For this, assume for simplicity that the first choice of π was v_1 and the output of D_{v_1} is indeed b_{v_1} (i.e., we are standing at point A on the tree depicted in Figure A.1). At this moment, the noise affects the deterministic outcome of v_1 (which we have assumed to be b_{v_1}) and hence π observes a noisy version of b_{v_1}. Based on this observation, π selects a new test (which might be a randomized selection). Let us assume that this time π selects the i-th test with probability $p_{1,i}^{\pi}$. An important point to note here is that conditioned on the fact that $D_{v_1} = b_{v_1}$ (i.e. point A on the tree), the new *choice* of π does not give any new information about Y. This is because conditioned on $D_{v_1} = b_{v_1}$, the choice of π is only a function of b_{v_1} and the noise and possibly some other random variables (used to randomise the policy) that are independent of Y given $D_{v_1} = b_{v_1}$. Hence, we can write

$$\mathbb{P}[\Lambda_2 \mid D_{v_1} = b_{v_1}] = \sum_{i=1}^{t} p_{1,i}^{\pi} \mathbb{P}[D_{v_i} = b_{v_i} \mid D_{v_1} = b_{v_1}]$$

Indeed, the above argument is valid if π had chosen any generic test v_i (instead of v_1) as its first test. The value $\mathbb{P}[\Lambda_2]$ can now be found by summing up the probabilities of

all the points at level 2 that are the end-point of a blue path. We have

$$\begin{aligned}
\mathbb{P}[\Lambda_2] &= \sum_{i=1}^{t} p_i^{\pi} \mathbb{P}[D_{v_i} = b_{v_i}] \mathbb{P}[\Lambda_2 \mid D_{v_i} = b_{v_i}] \\
&= \sum_{i,j=1}^{t} p_i^{\pi} p_{i,j}^{\pi} \mathbb{P}[D_{v_i} = b_{v_i}] \mathbb{P}\left[D_{v_j} = b_{v_j} \mid D_{v_i} = b_{v_i}\right] \\
&= \sum_{i,j=1}^{t} p_i^{\pi} p_{i,j}^{\pi} \mathbb{P}\left[D_{v_j} = b_{v_j}, D_{v_i} = b_{v_i}\right] \\
&\overset{(a)}{\geq} \sum_{i,j=1}^{t} p_i^{\pi} p_{i,j}^{\pi} (1 - 2(1-\beta)) \\
&\geq (1 - 2(1-\beta)) \sum_{i=1}^{t} p_i^{\pi} \sum_{j=1}^{t} p_{i,j}^{\pi} \\
&= 1 - 2(1-\beta),
\end{aligned}$$

where (a) follows from the Union bound.

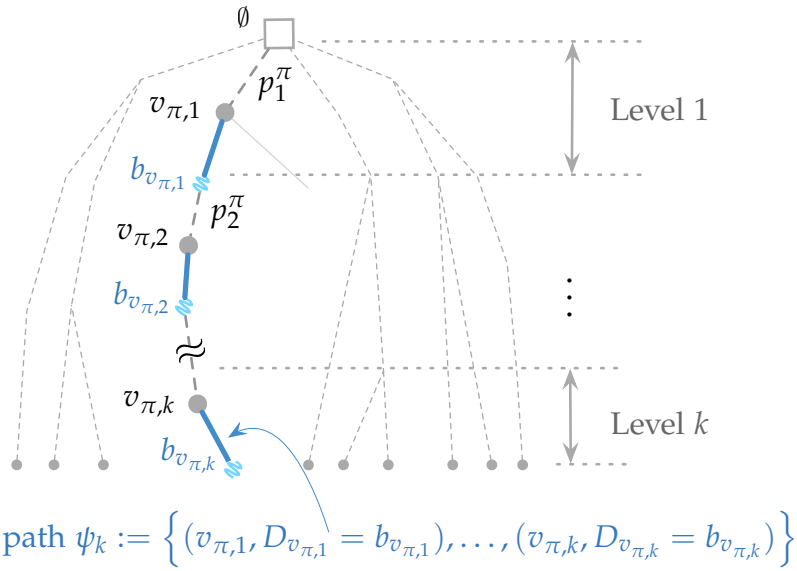


Figure A.2: Event Λ_k in the policy tree.

Now consider the general case where π has length k. As explained before, the event Λ happens only on the "good paths" (i.e., the paths that event Λ happens, as depicted in blue in Figure A.2) of the policy tree. Define path

$$\psi_i := \left\{(v_{\pi,1}, D_{v_{\pi,1}} = b_{v_{\pi,1}}), \ldots, (v_{\pi,i}, D_{v_{\pi,i}} = b_{v_{\pi,i}})\right\}.$$

We then have

$$\begin{aligned}
\mathbb{P}[\Lambda_k] &= \sum_{\psi_k} \mathbb{P}\left[(v_{\pi,1}, D_{v_{\pi,1}} = b_{v_{\pi,1}}), \ldots, (v_{\pi,k}, D_{v_{\pi,k}} = b_{v_{\pi,k}})\right] \\
&= \sum_{\psi_k} \mathbb{P}[v_{\pi,1}]\, \mathbb{P}\left[D_{v_{\pi,1}} = b_{v_{\pi,1}}\right] \prod_{i=1}^{k-1} \mathbb{P}[(]\, v_{\pi,i+1} \mid \psi_i) \times \\
&\qquad\qquad \prod_{i=1}^{k-1} \mathbb{P}\left[D_{v_{\pi,i+1}} = b_{v_{\pi,i+1}} \mid D_{v_{\pi,1}} = b_{v_{\pi,1}}, \ldots, D_{v_{\pi,i}} = b_{v_{\pi,i}}\right] \\
&= \sum_{\psi_k} \mathbb{P}\left[D_{v_{\pi,1}} = b_{v_{\pi,1}}, \ldots, D_{v_{\pi,k}} = b_{v_{\pi,k}}\right] \times \mathbb{P}[v_{\pi,1}] \prod_{i=1}^{k-1} \mathbb{P}[v_{\pi,i+1} \mid \psi_i]
\end{aligned}$$

Since for each $v_{\pi,i}$ it holds that $\mathbb{P}\left[D_{v_{\pi,i}} = b_{v_{\pi,i}}\right] \geq \beta$, applying the union bound we obtain

$$\mathbb{P}\left[D_{v_{\pi,1}} = b_{v_{\pi,1}}, \ldots, D_{v_{\pi,k}} = b_{v_{\pi,k}}\right] \geq 1 - k(1-\beta).$$

Thus,

$$\begin{aligned}
\mathbb{P}[\Lambda_k] &\geq (1 - k(1-\beta)) \sum_{\psi_k} \mathbb{P}[v_{\pi,1}] \prod_{i=1}^{k-1} \mathbb{P}[v_{\pi,i+1} \mid \psi_i] \\
&= (1 - k(1-\beta)) \sum_{\psi_{k-1}} \mathbb{P}[v_{\pi,1}] \prod_{i=1}^{k-2} \mathbb{P}[v_{\pi,i+1} \mid \psi_i] \sum_{v \in [t]} \mathbb{P}[v_{\pi,k} = v \mid \psi_{k-1}] \\
&= (1 - k(1-\beta)) \sum_{\psi_{k-2}} \mathbb{P}[v_{\pi,1}] \prod_{i=1}^{k-3} \mathbb{P}[v_{\pi,i+1} \mid \psi_i] \sum_{v \in [t]} \mathbb{P}[v_{\pi,k-1} = v \mid \psi_{k-2}] \\
&\quad \vdots \\
&= (1 - k(1-\beta)) \sum_{\psi_2} \mathbb{P}[v_{\pi,1}]\, \mathbb{P}[v_{\pi,2} \mid \psi_1] \sum_{v \in [t]} \mathbb{P}[v_{\pi,3} = v \mid \psi_2] \\
&= (1 - k(1-\beta)) \sum_{\psi_1} \mathbb{P}[v_{\pi,1}] \sum_{v \in [t]} \mathbb{P}[v_{\pi,2} = v \mid \psi_1] \\
&= (1 - k(1-\beta))
\end{aligned}$$

□

Lemma A.2. *Consider a distribution $p(\cdot)$ on set $\mathcal{U}'$ with $|\mathcal{U}'| = n$. For a subset $\mathcal{U} \subseteq \mathcal{U}'$ we have*

$$\sum_{u \in \mathcal{U}} p_u \log \frac{1}{p_u} \geq \mathbb{H}(p) - (1 - p(\mathcal{U})) \log n + (1 - p(\mathcal{U})) \log(1 - p(\mathcal{U})). \tag{A.1.3}$$

Proof. Let $\mathcal{U}^{\complement} \triangleq \mathcal{U}' \setminus \mathcal{U}$ be the complement of set $\mathcal{U}$. We have

$$\begin{aligned}
\mathbb{H}(p) - \sum_{u\in\mathcal{U}} p_u \log\frac{1}{p_u} &= \sum_{u\in\mathcal{U}^{\complement}} p_u \log\frac{1}{p_u} \\
&= p(\mathcal{U}^{\complement}) \sum_{u\in\mathcal{U}^{\complement}} \frac{p_u}{p(\mathcal{U}^{\complement})} \log\frac{p(\mathcal{U}^{\complement})}{p_u} - p(\mathcal{U}^{\complement}) \log p(\mathcal{U}^{\complement}) \\
&\overset{(a)}{\leq} p(\mathcal{U}^{\complement}) \log n - p(\mathcal{U}^{\complement}) \log p(\mathcal{U}^{\complement}) \\
&= (1 - p(\mathcal{U})) \log n - (1 - p(\mathcal{U})) \log(1 - p(\mathcal{U})),
\end{aligned}$$

where step (a) is due to the fact that the cardinality of the set $\mathcal{U}^{\complement}$ is at most n and thus the entropy of any distribution on this set is less than $\log n$. □

A.1.3 Proof of Lemma 3.5 for $n = 2$

For $n = 2$ we have $Y = \text{Bernoulli}(p)$. Assume w.l.o.g that $p \leq 1/2$. Each D_v is a deterministic function of Y. So D_v is itself a binary random variable. Now, there exists $v' \in [t]$ such that $\mathbb{I}(D_{v'}; Y) > 0$, otherwise any policy gains zero mutual information and the result of Lemma 3.5 is trivial. We assume w.l.o.g that $D_{v'} = Y$. By using part 2 of Lemma A.1 we get that $\mathbb{I}(X_{v'}; Y) \geq p\frac{S_{\min}}{2}$. Note that $H(Y) = \mathbb{H}_2(p)$, where $\mathbb{H}_2(x) \triangleq -x\log x - (1-x)\log(1-x)$. Also, it is easy to verify that for $p \leq 1/2$ we have $\mathbb{H}_2(p) \leq -2p\log p$ and also $-\log(\mathbb{H}_2(p)) \geq -\frac{\log p}{3}$. We thus get that

$$\mathbb{I}(X_{v'}; Y) \geq \frac{S_{\min}}{12} \frac{\mathbb{H}_2(p)}{\log 1/\mathbb{H}_2(p)}. \tag{A.1.4}$$

Now, note that any policy can have at most $\mathbb{I}(\pi, Y) \leq H(Y) = \mathbb{H}_2(p)$. Thus, $\log(1/\mathbb{I}(\pi, Y)) \geq \log(1/\mathbb{H}_2(p))$. As a result,

$$\frac{\mathbb{I}(\pi, Y)}{\log(1/\mathbb{I}(\pi, Y))} \leq \frac{\mathbb{H}_2(p)}{\log(1/\mathbb{H}_2(p))}. \tag{A.1.5}$$

To get the result of Lemma 3.5, we assume two cases: (i) $\mathbb{I}(\pi, Y) \leq \frac{1}{2}$: in this case $\log(1/\mathbb{I}(\pi, Y)) \geq 1$ and by (A.1.4) and (A.1.5) we obtain that $\mathbb{I}(X_{v'}; Y) \geq \frac{S_{\min}\mathbb{I}(\pi,Y)}{12\log(1/\mathbb{I}(\pi,Y))}$ (ii) $\mathbb{I}(\pi, Y) > \frac{1}{2}$ which, by using $\mathbb{H}_2(p) > \frac{1}{2}$, means that $p > 0.1102$ and thus $\frac{p}{\mathbb{H}_2(p)} \geq 1/6$. In this case, we have $\mathbb{I}(X_{v'}; Y) \geq p\frac{S_{\min}}{2} \geq \mathbb{H}_2(p) S_{\min}\frac{p}{2\mathbb{H}_2(p)} \geq \frac{S_{\min}\mathbb{H}_2(p)}{12} \geq \frac{S_{\min}\mathbb{I}(\pi;Y)}{12}$. Thus, form the two cases, we have proven that $\mathbb{I}(X_{v'}; Y) \geq \frac{S_{\min}\mathbb{I}(\pi;Y)}{12\max\{\log n, \log(1/\mathbb{I}(\pi,Y))\}}$ for any policy π. This proves Lemma 3.5 for $k \geq 2$. Note that the result of Lemma 3.5 is trivially valid when $k = 1$.

A.1.4 Proof of Lemma 3.10: W.h.p. π_{MIS} Picks Tests in $\mathcal{V}_0$

Proof of Lemma 3.10. The proof goes through the following steps:

Step 1. Consider 2T observable random variables $X_1^{(0)}, X_2^{(0)}, \dots X_{2T}^{(0)}$ from set $\mathbf{X}_{\mathcal{V}_0}$. If $S_{\min} = (1-2\epsilon)^2 \leq \frac{1}{256\sqrt{\log n}(\log\log n)^2}$, for all integers $r \leq 2T$, we claim

$$\mathbb{P}\left[\frac{1}{e} \leq \left(\frac{1-\epsilon}{\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}} \leq e\right] \geq 1 - 2\exp\left(-2(\log\log n)^2\right), \tag{A.1.6}$$

where $x_i^{(0)}$ is the observed outcome of $X_i^{(0)}$.

In the following, we prove the above inequality. By Hoeffding's inequality, we have

$$\mathbb{P}\left[\left|\sum_{i=1}^{r}(1-2x_i^{(0)}) - r(1-2\epsilon)\right| \geq \sqrt{r}\log\log n\right] \leq 2\exp\left(-2(\log\log n)^2\right)$$

Therefore, with probability at least $1-2\exp\left(-2(\log\log n)^2\right)$, we have

$$\begin{aligned}\left(\frac{1-\epsilon}{\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}} &\leq \left(\frac{1-\epsilon}{\epsilon}\right)^{r(1-2\epsilon)+\sqrt{r}\log\log n} \\ &= e^{\left(\ln\frac{1-\epsilon}{\epsilon}\cdot r\cdot(1-2\epsilon)+\sqrt{r}\ln\frac{1-\epsilon}{\epsilon}\cdot\log\log n\right)} \\ &\leq e^{\left(4(1-2\epsilon)^2 r+4\sqrt{r(1-2\epsilon)^2}\log\log n\right)} \\ &\leq e^{\left(4\times 2TS_{\min}+4\sqrt{2TS_{\min}}\log\log n\right)}\end{aligned}$$

In order for inequality $\left(\frac{1-\epsilon}{\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}} \leq e$ to hold, it suffices to ensure that

$$\begin{cases} 8TS_{\min} \leq \frac{1}{2} \\ 4\sqrt{2TS_{\min}}\log\log n \leq \frac{1}{2} \end{cases}$$

From the first inequality we get $S_{\min} \leq \frac{1}{32\sqrt{\log n}}$; from the second we get $S_{\min} \leq \frac{1}{256\sqrt{\log n}(\log\log n)^2}$.

To show $\left(\frac{1-\epsilon}{\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}} \geq \frac{1}{e}$, we use

$$\left(\frac{1-\epsilon}{\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}} \geq \left(\frac{1-\epsilon}{\epsilon}\right)^{r(1-2\epsilon)-\sqrt{r}\log\log n} = e^{\left(\ln\frac{1-\epsilon}{\epsilon}\cdot r\cdot(1-2\epsilon)-\sqrt{r}\ln\frac{1-\epsilon}{\epsilon}\cdot\log\log n\right)},$$

and it suffices to ensure that $\sqrt{r}\ln\frac{1-\epsilon}{\epsilon}\cdot\log\log n \le 1$, which clearly holds when $S_{\min} \le \frac{1}{256\sqrt{\log n}(\log\log n)^2}$. From (A.1.6) and the union bound we get that

$$\mathbb{P}\left[\forall r \le 2T : \frac{1}{e} \le \left(\frac{1-\epsilon}{\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}} \le e\right] \ge 1 - 4\sqrt{\log n}\exp\left(-2(\log\log n)^2\right)$$

Step 2. We now prove Lemma 3.10 by induction. Assume that $S_{\min} \ge \frac{2(10+2\log\log n)}{\log n}$. By equations (3.4.1),(3.4.2),(3.4.3), we know that the gain of any tests in $\mathcal{X}_t, t \in \{1,\dots,T+1\}$ is less that $S_{\min}$. In the very beginning, $\mathbb{I}\left(X_i^{(0)};Y\right) = (1-\mathbb{H}_2(\epsilon)) \ge S_{\min}$, so $\pi_{\mathrm{MIS}[2T]}$ chooses a test from $\mathbf{X}_{\mathcal{V}_0}$.

Step 3. Consider an integer $r \le 2T$ and assume that greedy has so far picked tests $X_1^{(0)},\dots,X_r^{(0)} \in \mathbf{X}_{\mathcal{V}_0}$ with outputs $x_1^{(0)},\dots,x_r^{(0)}$ such that $\frac{1}{e} \le \left(\frac{1-\epsilon}{\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}} \le e$.

We denote the probability of the event $y \in \mathcal{Y}^1$ by $p_1 = \mathbb{P}\left[y \in \mathcal{Y}^1 \mid x_1^{(0)},\dots,x_r^{(0)}\right]$, and similarly $p_0 = \mathbb{P}\left[y \in \mathcal{Y}^0 \mid x_1^{(0)},\dots,x_r^{(0)}\right]$. Then we have $p_1 + p_0 = 1$ and $\frac{p_1}{p_0} = \frac{\mathbb{P}\left[x_1^{(0)},\dots,x_r^{(0)}|y\in\mathcal{Y}^1\right]\mathbb{P}[y\in\mathcal{Y}^1]}{\mathbb{P}\left[x_1^{(0)},\dots,x_r^{(0)}|y\in\mathcal{Y}^0\right]\mathbb{P}[y\in\mathcal{Y}^0]} = \frac{(1-\epsilon)^{\sum_{i=1}^{r} x_i^{(0)}}\epsilon^{r-\sum_{i=1}^{r} x_i^{(0)}}}{\epsilon^{\sum_{i=1}^{r} x_i^{(0)}}(1-\epsilon)^{r-\sum_{i=1}^{r} x_i^{(0)}}} = \left(\frac{\epsilon}{1-\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}}$. Therefore, if $\frac{1}{e} \le \left(\frac{1-\epsilon}{\epsilon}\right)^{r-2\sum_{i=1}^{r} x_i^{(0)}} \le e$, then $p_1, p_0 \in [\frac{1}{4},\frac{3}{4}]$. Consider $X_i^{(0)} \in \mathbf{X}_{\mathcal{V}_0}$, and assume the distribution on D_i is a Bernoulli(p) with $p \in \left[\frac{1}{4},\frac{1}{2}\right]$, then

$$\begin{aligned}
\mathbb{I}\left(X_i^{(0)};Y\right) = \mathbb{I}\left(X_i^{(0)};D_i\right) &= \mathbb{H}_2\left(p(1-\epsilon)+(1-p)\epsilon\right) - \mathbb{H}_2(\epsilon) \\
&= \int_{\epsilon}^{p(1-\epsilon)+(1-p)\epsilon} \mathbb{H}_2'(x)\,dx \\
&\ge \frac{\mathbb{H}_2'(\epsilon)+\mathbb{H}_2'\left(p(1-\epsilon)+(1-p)\epsilon\right)}{2}\left(p(1-\epsilon)+(1-p)\epsilon-\epsilon\right) \\
&= p(1-2\epsilon)\frac{\log\frac{1-\epsilon}{\epsilon}+\log\frac{1-p(1-2\epsilon)-\epsilon}{p(1-2\epsilon)+\epsilon}}{2} \\
&\ge \frac{1-2\epsilon}{4}\frac{\log\frac{1-\epsilon}{\epsilon}+\log\left(1+\frac{2(1-2\epsilon)}{1+2\epsilon}\right)}{2} \\
&\ge \frac{1-2\epsilon}{8}\left(\frac{1-2\epsilon}{1-\epsilon}+(1-2\epsilon)\right) \\
&\ge \frac{3}{8}(1-2\epsilon)^2.
\end{aligned}$$

That is, when $p \in [1/4, 1/2]$, we have $\mathbb{I}\left(X_i^{(0)}; Y\right) \geq \frac{3}{8}(1-2\epsilon)^2 > \frac{1}{4}S_{\min} > \frac{10+2\log\log n}{2\log n}$.

Also, given the assumptions of step 3, we note that the elements in the set $\mathcal{Y}^1$ will always have equal probability. Therefore, the tests of other types have at most information twice as their gain in the very beginning where we have a uniform distribution on $\mathcal{Y}$. Therefore, the greedy policy will certainly choose a test among $\mathcal{V}_0$.

Finally, by combining Step 1-3, we finish the proof. □

A.2 Proofs from Chapter 4

A.2.1 Proof of Lemma 4.11: Adaptive Submodularity of f_{DiRECt}

In the following, we show that the function defined in Eq. (4.3.1) is strongly adaptive monotone and adaptive submodular.

Proof of Lemma 4.11. We first show f_{DiRECt} is strongly adaptively monotone: We know that each individual $f^i_{\text{EC}^2}$ is strongly adaptively monotone. Moreover, the partial derivative of f_{DiRECt} w.r.t. each $f^i_{\text{EC}^2}$ is non-negative. Applying the chain rule of derivatives, we know that f_{DiRECt} is strongly adaptively monotone.

To proof adaptive submodularity, we need to prove that for all $\mathbf{x}_\mathcal{A} \preceq \mathbf{x}_\mathcal{B}$ and $v \in \mathcal{V}$, it holds that $\Delta_{\text{DiRECt}}(v \mid \mathbf{x}_\mathcal{A}) \geq \Delta_{\text{DiRECt}}(v \mid \mathbf{x}_\mathcal{B})$. First, we introduce several auxiliary notations, as shown in Table A.1. Let $s_a(\mathbf{x}_\mathcal{A}) = \sum_i s_{i,a}(\mathbf{x}_\mathcal{A})$ be the number of hypotheses in the current hypotheses space given $\mathbf{x}_\mathcal{A}$ and $X_v = a$, and $s_\mathcal{V}(\mathbf{x}_\mathcal{A}) = |\mathcal{H}(\mathbf{x}_\mathcal{A})|$ be the number of hypotheses that are consistent with the observation $\mathbf{x}_\mathcal{A}$ (see Table A.1 for a list of notations used in this proof).

Table A.1: A reference table of auxiliary notations used for the proof of Lemma 4.11.

n	$\lvert\mathcal{H}(\emptyset)\rvert$, the number of hypotheses in the initial version space.
$s_\mathcal{V}(\mathbf{x}_\mathcal{A})$	$\lvert\mathcal{H}(\mathbf{x}_\mathcal{A})\rvert$, the number of hypotheses consistent with the observation $\mathbf{x}_\mathcal{A}$.
$s_i(\mathbf{x}_\mathcal{A})$	$\lvert\mathcal{H}(\mathbf{x}_\mathcal{A}) \cap \mathcal{R}_{y_i}\rvert$, the number of hypotheses in $\mathcal{R}_{y_i}$ that are consistent with $\mathbf{x}_\mathcal{A}$.
$s_a(\mathbf{x}_\mathcal{A})$	$\sum_i s_{i,a}(\mathbf{x}_\mathcal{A})$, the number of hypotheses in the current version space given $\mathbf{x}_\mathcal{A}$ and $X_v = a$.
$s_{i,a}(\mathbf{x}_\mathcal{A})$	$\lvert\{h : h \in \mathcal{H}(\mathbf{x}_\mathcal{A}, X_v = a) \cap \mathcal{R}_{y_i}\}\rvert$, the number of hypotheses in $\mathcal{R}_{y_i}$ that are consistent with the observation $\mathbf{x}_\mathcal{A}$ and $X_v = a$.
$\mathbf{s}(\mathbf{x}_\mathcal{A})$	the vector consisting of $s_{i,a}(\mathbf{x}_\mathcal{A})$ for all i and a.
ϕ	the expected marginal benefit of a test given some observations.

We can represent the marginal gain of f_{EC^2} on each graph as a function $\phi(\cdot)$ only

depending on $\mathbf{s}(\mathbf{x}_\mathcal{A})$:

$$\Delta_{f_{\text{EC}^2}}(v \mid \mathbf{x}_\mathcal{A}) = \phi(\mathbf{s}(\mathbf{x}_\mathcal{A})) = \frac{1}{2}\sum_{i\neq j}\sum_{a\neq b} s_{i,a}(\mathbf{x}_\mathcal{A}) \cdot s_{j,b}(\mathbf{x}_\mathcal{A}) + \sum_a \frac{s_a}{s_\mathcal{V}} \cdot \frac{1}{2}\sum_{i\neq j}\sum_{b\neq a} s_{i,b} \cdot s_{j,b} \tag{A.2.1}$$

Now let $s_{k,c}$ be the number of hypotheses in *auxiliary* equivalence class k, which are consistent with the observation $X_v = c$. Following the analysis of f_{EC^2} by Golovin, Krause, and Ray [GKR10a], we get $\partial\phi/\partial s_{k,c} \geq 0$ for any choice of k and c.

To show that $\Delta_{\text{DIRECT}}(v \mid \mathbf{x}_\mathcal{A}) = \phi_{f_{\text{DIRECT}}}(\mathbf{s}(\mathbf{x}_\mathcal{A}))$ is monotone decreasing with more observations, we need to show that for any k and c, it holds that $\partial\phi_{f_{\text{DIRECT}}}(\mathbf{s}(\mathbf{x}_\mathcal{A}))/\partial s_{k,c} \geq 0$. Denote the set $\mathcal{A} \cup \{v\}$ as $\mathcal{A} + v$. By the definition of $\Delta(v \mid \mathbf{x}_\mathcal{A})$, we know

$$\begin{aligned}
&\Delta_{\text{DIRECT}}(v \mid \mathbf{x}_\mathcal{A}) \\
=&\mathbb{E}\left[\left(1 - \prod_i^n \left(1 - f^i_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v})\right)\right) - \left(1 - \prod_i^n \left(1 - f^i_{\text{EC}^2}(\mathbf{x}_\mathcal{A})\right)\right)\right] \\
=&\mathbb{E}\left[(1 - f^1_{\text{EC}^2}(\mathbf{x}_\mathcal{A})) \cdot \prod_{i\neq 1}^n \left(1 - f^i_{\text{EC}^2}(\mathbf{x}_\mathcal{A})\right) - (1 - f^1_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v})) \cdot \prod_{i\neq 1}^n \left(1 - f^i_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v})\right)\right]
\end{aligned} \tag{A.2.2}$$

We first show for the simple case, where there are only two regions, the objective $f^{(2)}_{EC}$ is adaptive submodular w.r.t. uniform priors. For discussion simplicity we drop the normalization constants Q_i from the analysis.

Define $\delta_i(x_v \mid \mathbf{x}_\mathcal{A}) = f^i_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}) - f^i_{\text{EC}^2}(\mathbf{x}_\mathcal{A})$. If there are two regions, i.e., $n = 2$, Eq (A.2.2) becomes

$$\begin{aligned}
&\Delta_{\text{DIRECT}}(v \mid \mathbf{x}_\mathcal{A}) \\
=&\mathbb{E}\left[(1 - f^1_{\text{EC}^2}(\mathbf{x}_\mathcal{A})) \cdot (1 - f^2_{\text{EC}^2}(\mathbf{x}_\mathcal{A})) - (1 - f^1_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v})) \cdot (1 - f^2_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}))\right] \\
=&\mathbb{E}\Big[f^1_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}) - f^1_{\text{EC}^2}(\mathbf{x}_\mathcal{A}) + f^2_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}) - f^2_{\text{EC}^2}(\mathbf{x}_\mathcal{A}) \\
&\qquad - \left(f^1_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}) f^2_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}) - f^1_{\text{EC}^2}(\mathbf{x}_\mathcal{A}) f^2_{\text{EC}^2}(\mathbf{x}_\mathcal{A})\right)\Big] \\
=&\mathbb{E}\left[\delta_1(x_v \mid \mathbf{x}_\mathcal{A}) + \delta_2(x_v \mid \mathbf{x}_\mathcal{A}) - (\delta_1(x_v \mid \mathbf{x}_\mathcal{A}) f^2_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}) + \delta_2(x_v \mid \mathbf{x}_\mathcal{A}) f^1_{\text{EC}^2}(\mathbf{x}_\mathcal{A})) \mid \mathbf{x}_\mathcal{A}\right] \\
=&\mathbb{E}\left[(1 - f^1_{\text{EC}^2}(\mathbf{x}_\mathcal{A}))\delta_2(x_v \mid \mathbf{x}_\mathcal{A}) \mid \mathbf{x}_\mathcal{A}\right] + \mathbb{E}\left[(1 - f^2_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}))\delta_1(x_v \mid \mathbf{x}_\mathcal{A}) \mid \mathbf{x}_\mathcal{A}\right] \\
=&(1 - f^1_{\text{EC}^2}(\mathbf{x}_\mathcal{A}))\mathbb{E}\left[\delta_2(x_v \mid \mathbf{x}_\mathcal{A}) \mid \mathbf{x}_\mathcal{A}\right] + \mathbb{E}\left[(1 - f^2_{\text{EC}^2}(\mathbf{x}_{\mathcal{A}+v}))\delta_1(x_v \mid \mathbf{x}_\mathcal{A}) \mid \mathbf{x}_\mathcal{A}\right]
\end{aligned} \tag{A.2.3}$$

For the first term on the R.H.S. of Eq. (A.2.3), we have

$$\left(1-f^1_{\mathrm{EC}^2}(\mathbf{x}_{\mathcal{A}})\right)\mathbb{E}\left[\delta_2(x_v \mid \mathbf{x}_{\mathcal{A}}) \mid \mathbf{x}_{\mathcal{A}}\right] \geq \left(1-f^1_{\mathrm{EC}^2}(\mathbf{x}_{\mathcal{B}})\right)\mathbb{E}\left[\delta_2(x_v \mid \mathbf{x}_{\mathcal{B}}) \mid \mathbf{x}_{\mathcal{B}}\right] \tag{A.2.4}$$

Let the second term be $\theta(\mathbf{s})$, and denote $h(\mathbf{s}) = 1 - f^2_{\mathrm{EC}^2}(\mathbf{x}_{\mathcal{A}+v})$. In the following, we will show that $\partial\theta(\mathbf{s})/\partial s_{k,c} \geq 0$ for all $s_{k,c}$.

$$\begin{aligned}\theta(\mathbf{s}) &= \mathbb{E}\left[h(\mathbf{s})\delta_1(x_v \mid \mathbf{x}_{\mathcal{A}}) \mid \mathbf{x}_{\mathcal{A}}\right] \\ &= \sum_a h(\mathbf{s})\frac{s_a}{s_{\mathcal{V}}}\cdot\frac{1}{2}\left\{\sum_{i\neq j}\sum_{b\neq d} s_{i,b}(\mathbf{x}_{\mathcal{A}})\cdot s_{j,d}(\mathbf{x}_{\mathcal{A}}) + \sum_{i\neq j}\sum_{b\neq a} s_{i,b}(\mathbf{x}_{\mathcal{A}})\cdot s_{j,b}(\mathbf{x}_{\mathcal{A}})\right\}\end{aligned}$$

Taking the partial derivative of $\theta(\mathbf{s})$ w.r.t. $s_{k,c}$, we have

$$\begin{aligned}\frac{\partial\theta(\mathbf{s})}{\partial s_{k,c}} &= \sum_a \frac{\partial h(\mathbf{s})}{\partial s_{k,c}}\cdot\frac{s_a}{s_{\mathcal{V}}}\cdot\frac{1}{2}\left\{\sum_{i\neq j}\sum_{b\neq d} s_{i,b}(\mathbf{x}_{\mathcal{A}})\cdot s_{j,d}(\mathbf{x}_{\mathcal{A}}) + \sum_{i\neq j}\sum_{b\neq a} s_{i,b}(\mathbf{x}_{\mathcal{A}})\cdot s_{j,b}(\mathbf{x}_{\mathcal{A}})\right\} \\ &\quad + \sum_a h(\mathbf{s})\cdot\frac{\partial}{\partial s_{k,c}}\left\{\frac{s_a}{2n_{\mathcal{V}}}\cdot\sum_{i\neq j}\sum_{b\neq d} s_{i,b}(\mathbf{x}_{\mathcal{A}})\cdot s_{j,d}(\mathbf{x}_{\mathcal{A}}) + \frac{s_a}{2n_{\mathcal{V}}}\cdot\sum_{i\neq j}\sum_{b\neq a} s_{i,b}(\mathbf{x}_{\mathcal{A}})\cdot s_{j,b}(\mathbf{x}_{\mathcal{A}})\right\}\end{aligned} \tag{A.2.5}$$

Since $f^2_{\mathrm{EC}^2}(\mathbf{x}_{\mathcal{A}+v})$ is monotone decreasing w.r.t. $s_{k,c}$, $h(\mathbf{s})$ is monotone increasing, and thus $\partial h/\partial s_{k,c} \geq 0$. Therefore, the first term on the R.H.S. of Eq. (A.2.5) is nonnegative.

Let $p = \frac{1}{2}\sum_{i\neq j,b\neq d} s_{i,b}s_{j,d}$, and $q_a = \frac{1}{2}\sum_{i\neq j,b\neq a} s_{i,b}s_{j,b}$. For simplicity we drop the dependency of variables on $\mathbf{x}_{\mathcal{A}}$. Then the second term on the R.H.S. of Eq. (A.2.5) is

$$\begin{aligned}&\sum_a h(\mathbf{s})\cdot\frac{\partial}{\partial s_{k,c}}\left\{s_a\cdot\frac{1}{s_{\mathcal{V}}}\cdot p + s_a\cdot\frac{1}{s_{\mathcal{V}}}\cdot q_a\right\} \\ =&h(\mathbf{s})\cdot\underbrace{\frac{\partial}{\partial s_{k,c}}\left\{s_c\cdot\frac{1}{s_{\mathcal{V}}}\cdot p + s_c\cdot\frac{1}{s_{\mathcal{V}}}\cdot q_c\right\}}_{①} + \sum_{a\neq c} h(\mathbf{s})\cdot\underbrace{\frac{\partial}{\partial s_{k,c}}\left\{s_a\cdot\frac{1}{s_{\mathcal{V}}}\cdot p + s_a\cdot\frac{1}{s_{\mathcal{V}}}\cdot q_a\right\}}_{②}\end{aligned} \tag{A.2.6}$$

Expand term ① to get

$$\begin{aligned}① &= \frac{s_c}{s_{\mathcal{V}}}\cdot\frac{\partial p}{\partial s_{k,c}} + \frac{p}{s_{\mathcal{V}}}\cdot\cancelto{1}{\frac{\partial s_c}{\partial s_{k,c}}} + ps_c\cdot\frac{\partial(1/n_{\mathcal{V}})}{\partial s_{k,c}} + \frac{s_c}{s_{\mathcal{V}}}\cdot\cancelto{0}{\frac{\partial q_c}{\partial s_{k,c}}} + \frac{q_c}{s_{\mathcal{V}}}\cdot\cancelto{1}{\frac{\partial s_c}{\partial s_{k,c}}} + q_cs_c\cdot\frac{\partial(1/n_{\mathcal{V}})}{\partial s_{k,c}} \\ &= \frac{s_c}{s_{\mathcal{V}}}\cdot\sum_{j\neq k,b\neq c} s_{j,b} + \frac{p}{s_{\mathcal{V}}} - \frac{ps_c}{s^2_{\mathcal{V}}} + \frac{q_c}{s_{\mathcal{V}}} - \frac{q_cs_c}{s^2_{\mathcal{V}}} \\ &= \frac{s_c}{s_{\mathcal{V}}}\cdot\sum_{j\neq k,b\neq c} s_{j,b} + p\cdot\left(\frac{1}{s_{\mathcal{V}}} - \frac{s_c}{s^2_{\mathcal{V}}}\right) + q_c\cdot\left(\frac{1}{s_{\mathcal{V}}} - \frac{s_c}{s^2_{\mathcal{V}}}\right) \geq 0\end{aligned} \tag{A.2.7}$$

Similarly, for term ②,

$$
\begin{aligned}
② &= \frac{s_a}{s_{\mathcal{V}}} \cdot \underbrace{\frac{\partial p}{\partial s_{k,c}}}_{\sum_{j\neq k, b\neq c} s_{j,b}} + \frac{p}{s_{\mathcal{V}}} \cdot \cancelto{0}{\frac{\partial s_a}{\partial s_{k,c}}} + p s_a \cdot \frac{\partial (1/n_{\mathcal{V}})}{\partial s_{k,c}} + \frac{s_a}{s_{\mathcal{V}}} \cdot \underbrace{\frac{\partial q_a}{\partial s_{k,c}}}_{\sum_{j\neq k} s_{j,c}} + \frac{q_a}{s_{\mathcal{V}}} \cdot \cancelto{0}{\frac{\partial s_a}{\partial s_{k,c}}} + q_a s_a \cdot \frac{\partial (1/n_{\mathcal{V}})}{\partial s_{k,c}} \\
&= \frac{s_a}{s_{\mathcal{V}}} \cdot \sum_{j\neq k, b\neq c} s_{j,b} - \frac{p s_a}{s_{\mathcal{V}}^2} + \frac{s_a}{s_{\mathcal{V}}} \cdot \sum_{j\neq k} s_{j,c} - \frac{q_a s_a}{s_{\mathcal{V}}^2} \\
&= \frac{s_a}{s_{\mathcal{V}}} \cdot \left\{ \sum_{j\neq k} \sum_{b} s_{j,b} - \underbrace{\left(\frac{p}{s_{\mathcal{V}}} + \frac{q_a}{s_{\mathcal{V}}} \right)}_{③} \right\}
\end{aligned} \tag{A.2.8}
$$

Substitute $p = \frac{1}{2} \sum_{i\neq j, b\neq d} s_{i,b} s_{j,d}$, and $q_a = \frac{1}{2} \sum_{i\neq j, b\neq a} s_{i,b} s_{j,b}$ in term ③ to get:

$$
\begin{aligned}
\frac{p}{s_{\mathcal{V}}} + \frac{q_a}{s_{\mathcal{V}}} &= \frac{1}{2} \sum_{i\neq j, b\neq d} s_{i,b} \frac{s_{j,d}}{s_{\mathcal{V}}} + \frac{1}{2} \sum_{i\neq j, b\neq a} s_{i,b} \frac{s_{j,b}}{s_{\mathcal{V}}} \\
&\leq \frac{1}{s_{\mathcal{V}}} \cdot \frac{1}{2} \sum_{i\neq j, b\neq d} \left(s_{i,b} s_{j,d} + s_{i,b} s_{j,b} \right) \\
&\leq \frac{1}{s_{\mathcal{V}}} \left(\sum_{i,d} s_{i,d} \right) \cdot \left(\sum_{j\neq k} \sum_{b} s_{j,b} \right) \\
&= \sum_{j\neq k} \sum_{b} s_{j,b}
\end{aligned} \tag{A.2.9}
$$

Hence term ② is nonnegative. Combining Eq. (A.2.6) to A.2.9 with Eq. (A.2.5), we get $\partial\theta(\mathbf{s})/\partial s_{k,c} \geq 0$. Therefore, fix $\mathbf{x}_{\mathcal{A}} \preceq \mathbf{x}_{\mathcal{B}}$ and $v \in \mathcal{V}$, it holds that $\Delta_{\text{DiRECt}}(v \mid \mathbf{x}_{\mathcal{A}}) \geq \Delta_{\text{DiRECt}}(v \mid \mathbf{x}_{\mathcal{B}})$ for the case where there are two regions, and thus f_{EC^2} is adaptive submodular for $m = 2$ w.r.t. a uniform prior (note that we can adapt the proof technique from [GKR10a] to prove A.S. for arbitrary prior).

Now assume that $f^{(m)}_{\text{DiRECt}}$ is adaptive submodular for $m = k$ and $k > 2$, and we want to prove when $m = k + 1$, $f^{(k+1)}_{\text{DiRECt}}$ is also adaptive submodular. By definition, we have

$$
\begin{aligned}
f^{(k+1)}_{\text{DiRECt}} &= 1 - \prod_{i=1}^{k+1} \left(1 - f^{i}_{\text{EC}^2}(\mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}})) \right) \\
&= 1 - (1 - f^{k+1}_{\text{EC}^2}(\mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}})) \cdot \prod_{i=1}^{k} \left(1 - f^{i}_{\text{EC}^2}(\mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}})) \right) \\
&= 1 - (1 - f^{k+1}_{\text{EC}^2}(\mathcal{S}(\pi, \mathbf{x}_{\mathcal{V}})) \cdot (1 - f^{(k)}_{\text{DiRECt}})
\end{aligned}
$$

Since $f^{(k)}_{\text{DiRECt}}$ is adaptive submodular and strongly adaptive monotone, we can apply the same analysis for the two region case, to the above problem. Therefore, $f^{(k+1)}_{\text{DiRECt}}$ is adaptive submodular, and thus $f^{(n)}_{\text{DiRECt}}$ is adaptive submodular for any $n \geq 1$. □

Remarks ("intuitive explanation" of the proof). In fact, one can find concrete examples where Noisy-OR does not preserve adaptive submodularity. Fortunately, for EC^2-like objectives, we have proved that it does preserve adaptive submodularity. The intuition lies in that the EC^2 objective characterizes a class of adaptive submodular functions with certain structures, which offers enough slack for our proof to go through.

A.2.2 Proof of Theorem 4.12: Near-optimality of DiRECt

Proof. Let Q be the quota to be achieved, and η be any value such that $f_{\text{DiRECt}}(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) > Q - \eta$ implies $f_{\text{DiRECt}}(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) = Q$, then by Theorem 10 of [GK11a], the cost of π_{DiRECt} satisfies

$$\text{cost}_{avg}(\pi_{\text{DiRECt}}) \leq \text{cost}_{avg}(\pi^*)(\ln(Q/\eta) + 1).$$

In our case, apply $Q = 1$ and $\eta \geq \left(\frac{1}{p^2_{min}}\right)^n$ to get

$$\text{cost}_{avg}(\pi_{\text{DiRECt}}) \leq \text{cost}_{avg}(\pi^*)(2n \ln(1/p_{min}) + 1).$$

Hence it finishes the proof. □

A.2.3 Proof of Theorem 4.15: Upper Bounds on the Cost of $\pi^{g}_{\tilde{\mathcal{H}}}$

In this subsection, we provide proofs for the upper bounds on the cost of Algorithm 3. In the analysis, we assume that we only sample the hypotheses *once* at the beginning of each experiment (i.e., we don't resample after each iteration).

Proof. The main idea of the proof is illustrated in Fig. A.3.

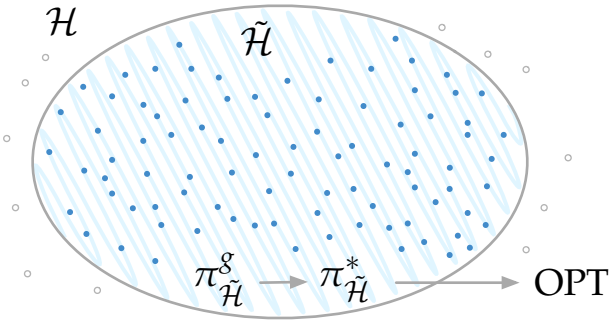


Figure A.3: Depicting the main idea behind the proof. We introduce $\pi^{*}_{\tilde{\mathcal{H}}}$ (the optimal policy on the sampled distribution) as an auxiliary policy to connect $\pi^{g}_{\tilde{\mathcal{H}}}$ with OPT. If the realized hypothesis $h^{*} \in \tilde{\mathcal{H}}$, then $\pi^{g}_{\tilde{\mathcal{H}}}$ efficiently identifies the decision. Otherwise, (with probability at most η) $\pi^{g}_{\tilde{\mathcal{H}}}$ randomly chooses tests, and the cost can be at most $C(\mathcal{V})$.

Bound on the Expected Cost

We first prove the upper bound on the expected cost of the algorithm. We use p to denote the real distribution over the hypotheses $h \in \mathcal{H}$, and $\tilde{p}$ be the sampled distribution. That is, $p(h) = \mathbb{P}[h]$, and

$$\tilde{p}(h) = \begin{cases} p(h)/(1-\eta), & \text{for } h \in \tilde{\mathcal{H}}; \\ 0, & \text{otherwise.} \end{cases} \tag{A.2.10}$$

For any policy π, let $\mathrm{cost}_{\tilde{p}}(\pi) \triangleq \mathbb{E}_{h\sim\tilde{p}(h)}[C(\mathcal{S}(\pi,h))]$ denote the expected cost of π w.r.t. $\tilde{p}$. To distinguish the expected cost of a policy w.r.t. the original distribution p from $\mathrm{cost}_{\tilde{p}}(\pi)$, in this subsection we use $\mathrm{cost}_{p}(\pi) \triangleq \mathrm{cost}_{avg}(\pi) = \mathbb{E}_{h\sim p(h)}[C(\mathcal{S}(\pi,h))]$.

The expected cost of π w.r.t. the true distribution p satisfies

$$\begin{aligned}
\mathrm{cost}_p(\pi) &= \sum_{h\in\mathcal{H}} p(h)C(\mathcal{S}(\pi,h)) \\
&= \sum_{h\in\tilde{\mathcal{H}}} p(h)C(\mathcal{S}(\pi,h)) + \sum_{h\in\mathcal{H}\setminus\tilde{\mathcal{H}}} p(h)C(\mathcal{S}(\pi,h)) \\
&\overset{\text{Eq. (A.2.10)}}{=} (1-\eta)\sum_{h\in\tilde{\mathcal{H}}} \tilde{p}(h)C(\mathcal{S}(\pi,h)) + \sum_{h\in\mathcal{H}\setminus\tilde{\mathcal{H}}} p(h)C(\mathcal{S}(\pi,h)) \\
&= (1-\eta)\,\mathrm{cost}_{\tilde{p}}(\pi) + \sum_{h\in\mathcal{H}\setminus\tilde{\mathcal{H}}} p(h)\underbrace{C(\mathcal{S}(\pi,h))}_{\leq C(\mathcal{V})} \qquad \text{(A.2.11)} \\
&\leq (1-\eta)\,\mathrm{cost}_{\tilde{p}}(\pi) + \eta\cdot C(\mathcal{V}). \qquad \text{(A.2.12)}
\end{aligned}$$

The second term on the RHS of Eq. (A.2.11) is non-negative, which gives

$$\begin{aligned}
(1-\eta)\,\mathrm{cost}_{\tilde{p}}(\pi) &= \mathrm{cost}_p(\pi) - \sum_{h\in\mathcal{H}\setminus\tilde{\mathcal{H}}} p(h)C(\mathcal{S}(\pi,h)) \\
&\leq \mathrm{cost}_p(\pi) \qquad \text{(A.2.13)}
\end{aligned}$$

Let $\pi^*_{\tilde{p}}$ be the optimal policy w.r.t. the sampled distribution $\tilde{p}$. By Theorem 2.7 we get

$$\mathrm{cost}_{\tilde{p}}\left(\pi^{g}_{\tilde{\mathcal{H}}}\right) \leq \left(r\ln\left(1/\tilde{p}_{\min}\right)+1\right)\mathrm{cost}_{\tilde{p}}\left(\pi^*_{\tilde{\mathcal{H}}}\right). \qquad \text{(A.2.14)}$$

Therefore,

$$\begin{aligned}
\mathrm{cost}_p(\pi^{g}_{\tilde{\mathcal{H}}}) &\overset{\text{Eq. (A.2.12)}}{\leq} (1-\eta)\,\mathrm{cost}_{\tilde{p}}(\pi^{g}_{\tilde{\mathcal{H}}}) + \eta\cdot C(\mathcal{V}) \\
&\overset{\text{Eq. (A.2.14)}}{\leq} (1-\eta)\left(r\ln\left(1/\tilde{p}_{\min}\right)+1\right)\mathrm{cost}_{\tilde{p}}\left(\pi^*_{\tilde{\mathcal{H}}}\right) \\
&\quad + \eta\cdot C(\mathcal{V}).
\end{aligned}$$

By definition, we know $\mathrm{cost}_{\tilde{p}}\left(\pi^*_{\tilde{\mathcal{H}}}\right) \leq \mathrm{cost}_{\tilde{p}}(\mathrm{OPT})$. Hence

$$\begin{aligned}
\mathrm{cost}_p(\pi^{g}_{\tilde{\mathcal{H}}}) &\leq (1-\eta)\left(r\ln\left(1/\tilde{p}_{\min}\right)+1\right)\mathrm{cost}_{\tilde{p}}(\mathrm{OPT}) + \eta\cdot C(\mathcal{V}) \\
&\overset{\text{Eq. (A.2.13)}}{\leq} \left(r\ln\left(1/\tilde{p}_{\min}\right)+1\right)\mathrm{cost}_p(\mathrm{OPT}) + \eta\cdot C(\mathcal{V}),
\end{aligned}$$

which completes the first part of the proof.

Bound on the Worst-case Cost

Next, we provide the proof for the bound on the worst-case cost. Analogous to the previous analysis, we consider two possible scenarios: (i) the realized hypotheses (i.e., the full realization vector) $h^* \in \tilde{\mathcal{H}}$; and (ii) $h^* \notin \tilde{\mathcal{H}}$.

For any policy π, the worst-case cost of π satisfies

$$\begin{aligned}\text{cost}_{wc}(\pi) &= \max_{h\in\mathcal{H}} C(\mathcal{S}(\pi,h)) \\ &= \max\{\max_{h\in\tilde{\mathcal{H}}} C(\mathcal{S}(\pi,h)), \max_{h\in\mathcal{H}\setminus\tilde{\mathcal{H}}} C(\mathcal{S}(\pi,h))\}.\end{aligned}$$

Since policy $\pi^g_{\tilde{\mathcal{H}}}$ terminates if there is no edge left on $\tilde{\mathcal{H}}$, then $\max_{h\in\mathcal{H}\setminus\tilde{\mathcal{H}}} C(\mathcal{S}(\pi,h)) \leq \max_{h\in\tilde{\mathcal{H}}} C(\mathcal{S}(\pi,h)$. Therefore,

$$\begin{aligned}\text{cost}_{wc}(\pi^g_{\tilde{\mathcal{H}}}) &= \max_{h\in\tilde{\mathcal{H}}} c\left(\mathcal{S}\left(\pi^g_{\tilde{\mathcal{H}}},h\right)\right) \\ &\overset{\text{(a)}}{\leq} (r\ln(1/\tilde{p}_{\min})+1)\max_{h\in\tilde{\mathcal{H}}} c\left(\mathcal{S}\left(\pi^*_{\tilde{\mathcal{H}}},h\right)\right) \\ &\leq (r\ln(1/\tilde{p}_{\min})+1)\max_{h\in H} c\left(\mathcal{S}\left(\text{OPT},h\right)\right).\end{aligned}$$

Step (a) in the above equation follows from Theorem A.12 of [GK11b].

Therefore, when $\pi^g_{\tilde{\mathcal{H}}}$ terminates, with probability at least $1-\eta$, it succeeds to output the correct decision with cost $(r\ln(1/\tilde{p}_{\min})+1)\,\text{cost}_{wc}(\text{OPT})$. □

A.2.4 Proof of Theorem 4.16: Lower Bound on the Value of $\pi^g_{\tilde{\mathcal{H}}}$

Assume that the cumulative probability of the enumerate hypotheses is at least $1-\eta$, i.e., using our sampling algorithm we enumerate $1-\eta$ fraction of the total mass.

Denote the set of sampled hypotheses by $\tilde{\mathcal{H}}$, and the expected gain of test v on $\tilde{\mathcal{H}}$ by $\Delta_{\tilde{\mathcal{H}}}(v \mid \cdot)$. Suppose we run the greedy algorithm based on $\tilde{\mathcal{H}}$. We want to show that the following lemma holds:

Lemma A.3. *Suppose $\tilde{\mathcal{H}} \subseteq \mathcal{H}$ and $\mathbb{P}\left[\tilde{\mathcal{H}}, \mathbf{x}_{\mathcal{A}}\right] \geq (1-\eta)\mathbb{P}\left[\mathcal{H}, \mathbf{x}_{\mathcal{A}}\right]$. Let $\tilde{v} \triangleq \arg\max_v \Delta_{\tilde{\mathcal{H}}}(v \mid \mathbf{x}_{\mathcal{A}})$ be the test with the maximal gain on $\tilde{\mathcal{H}}$ in the EC^2 objective function. Then for any test v, it holds that*

$$\Delta_{\mathcal{H}}(\tilde{v} \mid \mathbf{x}_{\mathcal{A}}) \geq \Delta_{\mathcal{H}}(v \mid \mathbf{x}_{\mathcal{A}}) - 2\eta\mathbb{P}\left[\mathbf{x}_{\mathcal{A}}\right]^2.$$

That is, the test $\tilde{v}$ which achieves the maximal gain on $\tilde{\mathcal{H}}$ will achieve a gain on $\mathcal{H}$ which is no less than $\varepsilon \triangleq 2\eta\mathbb{P}\left[\mathbf{x}_{\mathcal{A}}\right]^2$ below the maximal gain of any test. In the following, we provide the proof of Lemma A.3.

Proof. Clearly, if we can show that for any test v, the gain of v over $\tilde{\mathcal{H}}$ and the gain of v over $\mathcal{H}$ are at most ε apart, i.e.,

$$\Delta_{\mathcal{H}}(v \mid \mathbf{x}_{\mathcal{A}}) \leq \Delta_{\tilde{\mathcal{H}}}(v \mid \mathbf{x}_{\mathcal{A}}) + \varepsilon, \tag{A.2.15}$$

then we know that $\Delta_{\mathcal{H}}(v^* \mid \mathbf{x}_{\mathcal{A}}) \leq \Delta_{\tilde{\mathcal{H}}}(v^* \mid \mathbf{x}_{\mathcal{A}}) + \varepsilon \leq \Delta_{\tilde{\mathcal{H}}}(\tilde{v} \mid \mathbf{x}_{\mathcal{A}}) + \varepsilon$.

In the following, we show that inequality (A.2.15) holds.

The conditional expected gain of test v over observed tests $\mathbf{x}_{\mathcal{A}}$ is

$$\begin{aligned}\Delta_{\tilde{\mathcal{H}}}(v \mid \mathbf{x}_{\mathcal{A}}) &= \mathbb{E}[\delta_{\tilde{\mathcal{H}}}(x_v \mid \mathbf{x}_{\mathcal{A}})] \\ &= \mathbb{P}\left[x_v = 1 \mid \mathbf{x}_{\mathcal{A}}\right] \delta_{\tilde{\mathcal{H}}}(x_v = 1 \mid \mathbf{x}_{\mathcal{A}}) \\ &\quad + \mathbb{P}\left[x_v = 0 \mid \mathbf{x}_{\mathcal{A}}\right] \delta_{\tilde{\mathcal{H}}}(x_v = 0 \mid \mathbf{x}_{\mathcal{A}}).\end{aligned}$$

Here $\delta_{\tilde{\mathcal{H}}}(x_v \mid \mathbf{x}_{\mathcal{A}})$ denotes the conditional benefit of test v if its outcome is realized as x_v. Note that we can compute the probability terms $\mathbb{P}\left[x_v = 1 \mid \mathbf{x}_{\mathcal{A}}\right]$ and $\mathbb{P}\left[x_v = 0 \mid \mathbf{x}_{\mathcal{A}}\right]$ *exactly* from the CPT $\{\varrho_{ij}\}_{n\times m}$ via Bayesian update, i.e., $\mathbb{P}\left[x_v \mid \mathbf{x}_{\mathcal{A}}\right] = \sum_\theta \mathbb{P}\left[x_v, \theta \mid \mathbf{x}_{\mathcal{A}}\right] = \frac{\sum_\theta \mathbb{P}[\theta]\mathbb{P}[\mathbf{x}_{\mathcal{A}}|\theta]\mathbb{P}[x_v|\theta]}{\sum_\theta \mathbb{P}[\theta]\mathbb{P}[\mathbf{x}_{\mathcal{A}}|\theta]}$. What remains to be approximated is the gain for each specific

realization. For EC^2 object function, the gain of observing x_v over hypothesis set $\mathcal{H}$ after having observed $\mathbf{x}_\mathcal{A}$ is

$$\delta_\mathcal{H}(x_v \mid \mathbf{x}_\mathcal{A}) = \frac{1}{2}\sum_{y\neq y'}\left(\mathbb{P}\left[\mathcal{R}_y, \mathbf{x}_\mathcal{A}\right]\mathbb{P}\left[\mathcal{R}_{y'}, \mathbf{x}_\mathcal{A}\right] - \mathbb{P}\left[\mathcal{R}_y, \mathbf{x}_\mathcal{A}, x_v\right]\mathbb{P}\left[\mathcal{R}_{y'}, \mathbf{x}_\mathcal{A}, x_v\right]\right),$$

where $\mathcal{R}_y$ represents the set of hypotheses in *region / equivalence class* indexed by y. We define short-hand notation $\gamma_y := \mathbb{P}\left[\mathcal{R}_y \setminus \tilde{\mathcal{R}}_y, \mathbf{x}_A\right]$, where $\tilde{\mathcal{R}}_y$ denotes the sampled hypotheses within decision region $\mathcal{R}_y$. The difference in the gain of v over $\mathcal{H}$ and $\tilde{\mathcal{H}}$ can be expressed as

$$\begin{aligned}
&\delta_\mathcal{H}(x_v \mid \mathbf{x}_\mathcal{A}) - \delta_{\tilde{\mathcal{H}}}(x_v \mid \mathbf{x}_\mathcal{A}) \\
&\quad= \frac{1}{2}\sum_{y\neq y'}\left(\mathbb{P}\left[\mathcal{R}_y, \mathbf{x}_\mathcal{A}\right]\mathbb{P}\left[\mathcal{R}_{y'}, \mathbf{x}_\mathcal{A}\right] - \mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}\right]\mathbb{P}\left[\tilde{\mathcal{R}}_{y'}, \mathbf{x}_\mathcal{A}\right]\right) - \\
&\qquad\frac{1}{2}\sum_{y\neq y'}\left(\mathbb{P}\left[\mathcal{R}_y, \mathbf{x}_\mathcal{A}, x_v\right]\mathbb{P}\left[\mathcal{R}_{y'}, \mathbf{x}_\mathcal{A}, x_v\right] - \mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}, x_v\right]\mathbb{P}\left[\tilde{\mathcal{R}}_{y'}, \mathbf{x}_\mathcal{A}, x_v\right]\right) \\
&\quad\leq \frac{1}{2}\sum_{y\neq y'}\left(\mathbb{P}\left[\mathcal{R}_y, \mathbf{x}_\mathcal{A}\right]\mathbb{P}\left[\mathcal{R}_{y'}, \mathbf{x}_\mathcal{A}\right] - \mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}\right]\mathbb{P}\left[\tilde{\mathcal{R}}_{y'}, \mathbf{x}_\mathcal{A}\right]\right) \\
&\quad= \frac{1}{2}\sum_{y\neq y'}\left(\left(\mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}\right] + \gamma_y\right)\left(\mathbb{P}\left[\tilde{\mathcal{R}}_{y'}, \mathbf{x}_\mathcal{A}\right] + \gamma_{y'}\right) - \mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}\right]\mathbb{P}\left[\tilde{\mathcal{R}}_{y'}, \mathbf{x}_\mathcal{A}\right]\right) \\
&\quad= \frac{1}{2}\sum_{y\neq y'}\left(\gamma_y\left(\gamma_{y'} + \mathbb{P}\left[\tilde{\mathcal{R}}_{y'}, \mathbf{x}_\mathcal{A}\right]\right) + \gamma_{y'}\mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}\right]\right) \\
&\quad= \frac{1}{2}\sum_{y\neq y'}\left(\gamma_y\mathbb{P}\left[\mathcal{R}_{y'}, \mathbf{x}_\mathcal{A}\right] + \gamma_{y'}\mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}\right]\right) \\
&\quad\leq \sum_y \gamma_y \sum_{y'}\mathbb{P}\left[\mathcal{R}_{y'}, \mathbf{x}_\mathcal{A}\right] + \sum_{y'}\gamma_{y'}\sum_y \mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}\right].
\end{aligned}$$

By the definition of γ_y we know that

$$\sum_y \gamma_y = \mathbb{P}\left[\bigcup_i \left(\mathcal{R}_y \setminus \tilde{\mathcal{R}}_y\right), \mathbf{x}_\mathcal{A}\right] \overset{(a)}{=} \mathbb{P}\left[\left(\bigcup_i \mathcal{R}_y \setminus \bigcup_i \tilde{\mathcal{R}}_y\right), \mathbf{x}_\mathcal{A}\right] = \mathbb{P}\left[\mathcal{H} \setminus \tilde{\mathcal{H}}, \mathbf{x}_\mathcal{A}\right] \leq \eta\mathbb{P}\left[\mathbf{x}_\mathcal{A}\right].$$

Step (a) is because of the assumption that $\mathcal{R}_y$'s do not overlap. Hence,

$$\begin{aligned}
\Delta_\mathcal{H}(v \mid \mathbf{x}_\mathcal{A}) - \Delta_{\tilde{\mathcal{H}}}(v \mid \mathbf{x}_\mathcal{A}) &= \mathbb{E}[\delta_{\tilde{\mathcal{H}}}(x_v \mid \mathbf{x}_\mathcal{A})] \\
&\leq \eta\mathbb{P}\left[\mathbf{x}_\mathcal{A}\right]\sum_{y'}\mathbb{P}\left[\mathcal{R}_{y'}, \mathbf{x}_\mathcal{A}\right] + \eta\mathbb{P}\left[\mathbf{x}_\mathcal{A}\right]\sum_y \mathbb{P}\left[\tilde{\mathcal{R}}_y, \mathbf{x}_\mathcal{A}\right] \\
&\leq 2\eta\mathbb{P}\left[\mathbf{x}_\mathcal{A}\right]^2. \qquad \text{(A.2.16)}
\end{aligned}$$

Combining Equation (A.2.15) and (A.2.16) we finish the proof. □

Next, we provide the proof of Theorem 4.16 using the Lemma A.3.

Proof of Theorem 4.16. The key of the proof is to bound the one-step gain of the policy $\pi^g_{\tilde{\mathcal{H}},[\ell]}$.

$$
\begin{aligned}
& F(\pi^g_{\tilde{\mathcal{H}},[i+1]}) - F(\pi^g_{\tilde{\mathcal{H}},[i]}) \\
& \overset{\text{Lemma A.3}}{\geq} \mathbb{E}\left[\max_v(\Delta(v \mid \mathbf{x}_{\mathcal{A}})) - 2\eta\right] \\
& \overset{\text{(a)}}{\geq} \mathbb{E}\left[\frac{\Delta(\pi^*_{\mathcal{H},[k]} \mid \mathbf{x}_{\mathcal{A}})}{k} - 2\eta\right] \\
& = \mathbb{E}\left[\frac{F(\pi^*_{\mathcal{H},[k]}@\pi^g_{\tilde{\mathcal{H}},[i]}) - F(\pi^g_{\tilde{\mathcal{H}},[i]})}{k} - 2\eta\right] \\
& \overset{\text{(b)}}{\geq} \mathbb{E}\left[\frac{F(\pi^*_{\mathcal{H},[k]}) - F(\pi^g_{\tilde{\mathcal{H}},[i]})}{k} - 2\eta\right].
\end{aligned}
$$

Here $\pi^*_{\mathcal{H},[k]}@\pi^g_{\tilde{\mathcal{H}},[i]}$ denotes the concatenated policy of $\pi^*_{\mathcal{H},[k]}$ and $\pi^g_{\tilde{\mathcal{H}},[i]}$ (i.e., we first run $\pi^g_{\tilde{\mathcal{H}},[i]}$, and then run $\pi^*_{\mathcal{H},[k]}$ from scratch, ignoring the observations made by $\pi^g_{\tilde{\mathcal{H}},[i]}$).
The proof structure follows closely from the proof of Theorem A.10 in [GK11b]: Step (a) follows from the adaptive submodularity of f, and step (b) is due to monotonicity of F. Define $\Delta_i := F(\pi^*_{\mathcal{H},[k]}) - F(\pi^g_{\tilde{\mathcal{H}},[i]})$, from the above equation we get $\Delta_\ell \leq \left(1 - \frac{1}{k}\right)^l \Delta_0 + \sum_{i=0}^{l} \left(1 - \frac{1}{k}\right)^i$. Hence, $F\left(\pi^g_{\tilde{\mathcal{H}},[\ell]}\right) \geq \left(1 - e^{-\ell/k}\right) F\left(\pi^*_{\mathcal{H},[k]}\right) - 2k\eta\left(1 - \left(\frac{1}{k}\right)^\ell\right)$. □

A.2.5 Proof of Theorem 4.19 Outline: Introducing the Auxiliary Functions

In this subsection, we provide the proofs of our theoretical results in full detail. Recall that for the theoretical analysis, we study the basic setting where test outcomes are *binary*, and the test noise is *independent* of the underlying root-causes (i.e., given a test v, the noise rate on the outcome of test v is only a function of v, but not a function of θ).

The general idea behind our analysis, is to show that by running ECED, the one-step gain in learning the value of the target variable is significant, compared with the cumulative gain of an optimal policy over k steps (see Fig. A.4).

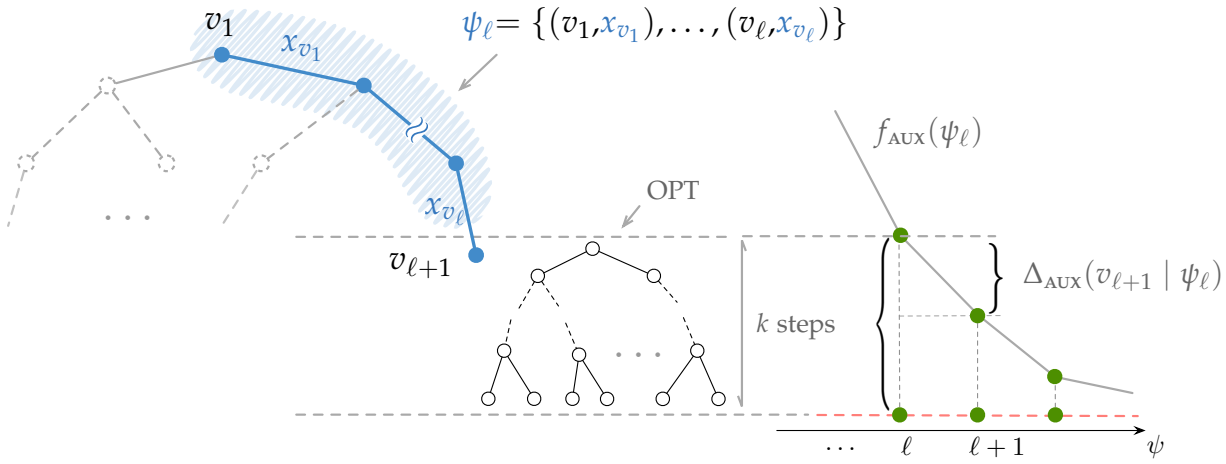


Figure A.4: On the left, we demonstrate a sequential policy in the form of its decision tree representation. Nodes represent tests selected by the policy, and edges represent outcomes of tests. At step ℓ, a policy maps partial realization $\psi_\ell = \{(v_1, x_{v_1}), \ldots, (v_\ell, x_{v_\ell})\}$ to the next test $v_{\ell+1}$ to be performed. In the middle, we demonstrate the tests selected by an optimal policy OPT of length k. On the right, we illustrate the change in the auxiliary function as ECED selects more tests. Running OPT at any step of execution of ECED will make f_{AUX} below some threshold (represented by the red dotted line). The key idea behind our proof, is to show that the greedy policy ECED, at each step, is making effective progress in reducing the expected prediction error (in the long run), compared with OPT.

In §4.2.1, we show that greedily optimizing a posterior-based objective function (e.g., expected prediction error) may perform arbitrarily poorly. In those cases, we may end up failing to pick some tests, which have negligible immediate gain regarding error

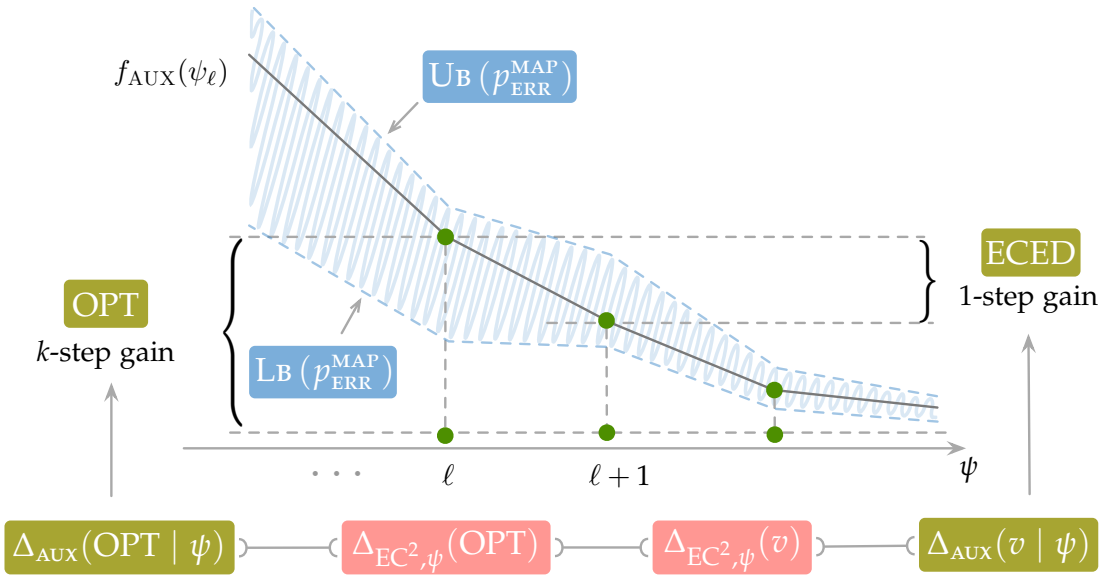


Figure A.5: The proof outline for Theorem 4.19.

reduction, but are very informative in the long run. ECED bypasses such an issue by selecting tests that maximally distinguish root-causes with different target values. In order to analyze ECED, we need to find an auxiliary function that properly tracks the "progress" of the ECED algorithm; meanwhile, this auxiliary function should allow us to connect the heuristic by which we select tests (i.e., Δ_{ECED}), with the target objective of interest (i.e., the expected prediction error p_{ERR}).

We consider the auxiliary function defined in Equation (4.5.3). For brevity, we suppress the dependence of ψ where it is unambiguous. Further, we use p_θ, $p_{\theta'}$, and p_y as shorthand notations for $\mathbb{P}[\theta \mid \psi]$, $\mathbb{P}[\theta' \mid \psi]$ and $\mathbb{P}[y \mid \psi]$. Equation (4.5.3) can be simplified as

$$f_{\text{AUX}} = \sum_{(\theta,\theta')\in\mathcal{E}} p_\theta p_{\theta'} \log \frac{1}{p_\theta p_{\theta'}} + c \sum_{y\in\mathcal{Y}} \mathbb{H}_2(p_y) \tag{A.2.17}$$

We illustrate the outline of our proofs in Fig. A.5. Our goal is to bound the cost of ECED against the cost of OPT (Theorem 4.19). As we have explained earlier, our strategy is to relate the one-step gain of ECED (i.e., $\Delta_{\text{AUX}}(v \mid \psi)$), with the gain of OPT in k-steps (i.e., $\Delta_{\text{AUX}}(\text{OPT} \mid \psi)$). To achieve that, we divide our proof into three parts:

Part 1 We show that the auxiliary function f_{AUX} is closely related with the target objective function p_{ERR}. More specifically, we provide both an upper bound $\text{UB}\,(p_{\text{ERR}}^{\text{MAP}})$ and

a lower bound $\textsc{Lb}\left(p_{\text{ERR}}^{\text{MAP}}\right)$ of f_{AUX} in Lemma 4.20, and give the detailed proofs in Appendix §A.2.6.

Part 2 To analyze the one-step gain of ECED, we introduce another intermediate auxiliary function: For a test $v_{\ell+1}$ chosen by ECED, we relate its one-step gain in the auxiliary function $\Delta_{\text{AUX}}(v \mid \psi)$, to its one-step gain in the EC^2 objective $\Delta_{\text{EC}^2,\psi}(v)$ (Lemma 4.21, detailed proof provided in Appendix §A.2.7). The reason why we introduce this step is that the EC^2 objective is adaptive submodular, by which we can relate the 1-step gain of a greedy policy $\Delta_{\text{EC}^2,\psi}(v)$ to an optimal policy $\Delta_{\text{EC}^2,\psi}(\text{OPT})$.

Part 3 To close the loop, it remains to connect the gain of an optimal policy OPT in the EC^2 objective function $\Delta_{\text{EC}^2,\psi}(\text{OPT})$, with the gain of OPT in the auxiliary function $\Delta_{\text{AUX}}(\text{OPT} \mid \psi)$. We establish such connection in Lemma A.8, and present its proof in Appendix §A.2.8.

To make the proof more accessible, we insert the annotated color blocks from Fig. A.5 (i.e., $\textsc{Ub}\left(p_{\text{ERR}}^{\text{MAP}}\right)$, $\textsc{Lb}\left(p_{\text{ERR}}^{\text{MAP}}\right)$, $\Delta_{\text{AUX}}(v \mid \psi)$, $\Delta_{\text{EC}^2,\psi}(v)$, $\Delta_{\text{EC}^2,\psi}(\text{OPT})$, $\Delta_{\text{AUX}}(\text{OPT} \mid \psi)$, etc), into the subsequent subsections, so that readers can easily relate different parts of this section to the proof outline. Note that we only use these annotated color blocks for positioning the proofs, and hence readers can ignore the notations, as it may slightly differ from the ones used in the proof.

A.2.6 Proof of Theorem 4.19 Part 1: Proof of Lemma 4.20

In this subsection, we provide the proof of Lemma 4.20, which relates f_{AUX} to p_{ERR}.

Define $p_{\text{E}}(\psi) \triangleq \sum_{y \in \mathcal{Y}} \mathbb{P}[y \mid \psi]\left(1 - \mathbb{P}[y \mid \psi]\right)$ as the prediction error of a *stochastic estimator* upon observing ψ, i.e., the probability of mispredicting y if we make a random draw from $\mathbb{P}[Y \mid \psi]$. We show in Lemma A.4 that $p_{\text{ERR}}^{\text{MAP}}(\psi)$ is within a constant factor of $p_{\text{E}}(\psi)$:

Lemma A.4. *Fix ψ, it holds that $p_{\text{ERR}}^{\text{MAP}}(\psi) \leq p_{\text{E}}(\psi) \leq 2p_{\text{ERR}}^{\text{MAP}}(\psi)$.*

Proof of Lemma A.4. We can always lower bound p_{E} by $p_{\text{ERR}}^{\text{MAP}}$, since by definition, $p_{\text{ERR}}^{\text{MAP}}(\psi) = 1 - \max_y \mathbb{P}[y \mid \psi] = \sum_{y \in \mathcal{Y}} \mathbb{P}[y \mid \psi] \cdot \left(1 - \max_y \mathbb{P}[y \mid \psi]\right) \leq \sum_{y \in \mathcal{Y}} \mathbb{P}[y \mid \psi]\left(1 - \mathbb{P}[y \mid \psi\right.$ $p_{\text{E}}(\psi)$.

To prove the second part, we write $p_{y_i} = \mathbb{P}[Y = y_i \mid \psi]$ for all $y_i \in \mathcal{Y}$. W.l.o.g., we assume $p_{y_1} \geq p_{y_2} \geq \cdots \geq p_{y_n}$. Then $p_{\text{ERR}}^{\text{MAP}} = 1 - p_{y_1}$. We further have

$$\begin{aligned}
2p_{\text{ERR}}^{\text{MAP}} &= 2(1 - p_{y_1}) = 2(\sum_{i=2}^{n} p_{y_i}) = 2(\sum_{i=1}^{n} p_{y_i})(\sum_{i=2}^{n} p_{y_i}) = 2(p_{y_1} + \sum_{i=2}^{n} p_{y_i})(\sum_{i=2}^{n} p_{y_i}) \\
&\geq 2p_{y_1}(\sum_{i=2}^{n} p_{y_i}) + (\sum_{i=2}^{n} p_{y_i})^2 \\
&\geq \sum_{i \neq j}^{n} p_{y_i} p_{y_j} = \sum_{i} p_{y_i}(1 - p_{y_i}) = p_{\text{E}}
\end{aligned}$$

□

Now, we provide lower and upper bounds of the second term in the RHS of Equation (A.2.17):

Lemma A.5. $2p_{\text{ERR}}^{\text{MAP}} \leq \sum_{y \in \mathcal{Y}} \mathbb{H}_2(p_y) \leq 3(\mathbb{H}_2(p_{\text{ERR}}^{\text{MAP}}) + p_{\text{ERR}}^{\text{MAP}} \log m)$.

Proof of Lemma A.5. We first prove the inequality on the left. Expanding the middle term involving the binary entropy of p_y, we get

$$\begin{aligned}
\sum_{y \in \mathcal{Y}} \mathbb{H}_2(p_y) &= \sum_{y \in \mathcal{Y}} \left(p_y \log \frac{1}{p_y} + (1 - p_y) \log \frac{1}{1 - p_y} \right) \\
&\overset{\text{(a)}}{\geq} \frac{2}{\ln 2} \sum_{y \in \mathcal{Y}} p_y(1 - p_y) \\
&\geq 2p_{\text{E}} \overset{\text{Lemma A.4}}{\geq} 2p_{\text{ERR}}^{\text{MAP}}
\end{aligned}$$

Here, step (a) is by inequality $\ln x \geq 1 - 1/x$ for $x \geq 0$.

To prove the second part, we first show in the following that $\sum_y (1 - p_y) \log \frac{1}{1-p_y} \leq 2 \sum_y p_y \log \frac{1}{p_y}$.

W.l.o.g., we assume that the probabilities p_y's are in decreasing order, i.e., $p_{y_1} \geq p_{y_2} \geq \cdots \geq p_{y_n}$. Observe that if $p_y \in [0, 1/2]$, then $(1 - p_y) \log \frac{1}{1-p_y} \leq p_y \log \frac{1}{p_y}$. Consider the following two cases:

1. $p_{y_1} \leq 1/2$. In this case, we have $\sum_y (1 - p_y) \log \frac{1}{1-p_y} \leq \sum_y p_y \log \frac{1}{p_y}$.

2. $p_{y_1} > 1/2$. Since $\sum_{i>1} p_{y_i} = 1 - p_{y_1}$, we have

$$\begin{aligned}
\sum_i (1-p_{y_i}) \log \frac{1}{1-p_{y_i}} &= (1-p_{y_1}) \log \frac{1}{1-p_{y_1}} + \sum_{i>1} (1-p_{y_i}) \log \frac{1}{1-p_{y_i}} \\
&= \sum_{i>1} p_{y_i} \log \frac{1}{\sum_{i>1} p_{y_i}} + \sum_{i>1} (1-p_{y_i}) \log \frac{1}{1-p_{y_i}} \\
&\leq \sum_{i>1} p_{y_i} \log \frac{1}{p_{y_i}} + \sum_{i>1} (1-p_{y_i}) \log \frac{1}{1-p_{y_i}} \\
&\leq \sum_{i>1} p_{y_i} \log \frac{1}{p_{y_i}} + \sum_{i>1} p_{y_i} \log \frac{1}{p_{y_i}} \\
&\leq 2 \sum_{i>0} p_{y_i} \log \frac{1}{p_{y_i}}
\end{aligned}$$

Therefore,

$$\sum_{y \in \mathcal{Y}} \mathbb{H}_2 \left(p_y \right) \leq 3 \sum_{i>0} p_{y_i} \log \frac{1}{p_{y_i}} = 3 \mathbb{H} \left(Y \right). \tag{A.2.18}$$

Furthermore, by Fano's inequality (in the absence of conditioning), we know that $\mathbb{H}(Y) \leq \mathbb{H}_2 \left(p_{\text{ERR}}^{\text{MAP}} \right) + p_{\text{ERR}}^{\text{MAP}} \log(|\mathcal{Y}| - 1)$. Combining with Equation (A.2.18) we get

$$\sum_y \mathbb{H}_2 \left(p_y \right) \leq 3 \mathbb{H} \left(Y \right) \leq 3 \left(\mathbb{H}_2 \left(p_{\text{ERR}}^{\text{MAP}} \right) + \log(|\mathcal{Y}| - 1) \right) \overset{(b)}{\leq} 3 \left(\mathbb{H}_2 \left(p_{\text{ERR}}^{\text{MAP}} \right) + \log m \right)$$

where in (b) we use the fact that $t = |\mathcal{Y}| \leq |\operatorname{supp}(\Theta)| = m$, since $Y = r(\Theta)$ is a function of Θ. Hence it completes the proof. □

Next, we bound the first term on the RHS of Equation (A.2.17), i.e., $\sum_{\{\theta,\theta'\} \in \mathcal{E}} p_\theta p_{\theta'} \log \frac{1}{p_\theta p_{\theta'}}$, against $p_{\text{ERR}}^{\text{MAP}}$:

Lemma A.6. $\sum_{\{\theta,\theta'\} \in \mathcal{E}} p_\theta p_{\theta'} \log \frac{1}{p_\theta p_{\theta'}} \leq 2(\mathbb{H}_2 \left(p_{\text{E}} \right) + p_{\text{E}} \log m)$.

Proof of Lemma A.6. We can expand the LHS as

$$\begin{aligned}
\text{LHS} &= -\sum_{\theta'} p_{\theta'} \sum_{\theta: r(\theta)\neq r(\theta')} p_\theta (\log p_\theta + \log p_{\theta'}) \\
&= -2\sum_{\theta'} p_{\theta'} \sum_{\theta: r(\theta)\neq r(\theta')} p_\theta \log p_\theta \\
&= -2\sum_{y\in\mathcal{Y}} \sum_{\theta': r(\theta')=y} p_{\theta'} \sum_{\theta: r(\theta)\neq y} p_\theta \log p_\theta \\
&= 2\sum_{y\in\mathcal{Y}} p_y(1-p_y) \sum_{\theta: r(\theta)\neq y} \frac{p_\theta}{1-p_y} \left(\log \frac{p_\theta}{1-p_y} + \log(1-p_y) \right) \\
&= -2\sum_{y\in\mathcal{Y}} p_y(1-p_y)\log(1-p_y) + 2\sum_{y\in\mathcal{Y}} p_y(1-p_y)\mathbb{H}\left(\left\{ \frac{p_\theta}{(1-p_y)} \right\}_{\theta: r(\theta)\neq y} \right) \qquad \text{(A.2.19)} \\
&\leq 2\sum_{y\in\mathcal{Y}} p_y \mathbb{H}_2(1-p_y) + 2\sum_{y\in\mathcal{Y}} p_y(1-p_y)\mathbb{H}\left(\left\{ \frac{p_\theta}{(1-p_y)} \right\}_{\theta: r(\theta)\neq y} \right)
\end{aligned}$$

Since $\mathbb{H}\left(\left\{ \frac{p_\theta}{(1-p_y)} \right\}_{\theta: r(\theta)\neq y} \right) \leq \log n \leq \log m$, we have

$$\begin{aligned}
\text{LHS} &\leq 2\sum_{y\in\mathcal{Y}} p_y \mathbb{H}_2(1-p_y) + 2\sum_{y} \underbrace{p_y(1-p_y)\log m}_{p_{\text{E}}\log m} \\
&\overset{\text{Jensen}}{\leq} 2\mathbb{H}_2\left(\sum_{y\in\mathcal{Y}} p_y(1-p_y) \right) + 2p_{\text{E}}\log m \\
&= 2\left(\mathbb{H}_2(p_{\text{E}}) + p_{\text{E}}\log m \right).
\end{aligned}$$

which completes the proof. □

Now, we are ready to state the upper bound $\textsc{Ub}\left(p_{\text{ERR}}^{\text{MAP}}\right)$ and lower bound $\textsc{Lb}\left(p_{\text{ERR}}^{\text{MAP}}\right)$ of f_{AUX}.

Proof of Lemma 4.20. Clearly, $\sum_{\{\theta,\theta'\}\in\mathcal{E}} p_\theta p_{\theta'} \log \frac{1}{p_\theta p_{\theta'}} \geq 0$. By Lemma A.5 we get the lower bound:

$$f_{\text{AUX}}(\psi) \geq 2c \cdot p_{\text{ERR}}^{\text{MAP}}(\psi).$$

Now assume $p_{\text{ERR}}^{\text{MAP}} \leq 1/4$. By Lemma A.4 we know $p_{\text{E}} \leq 2p_{\text{ERR}}^{\text{MAP}}$, and $\mathbb{H}_2(p_{\text{E}}) \leq \mathbb{H}_2(2p_{\text{ERR}}^{\text{MAP}}) \leq 2\mathbb{H}_2(p_{\text{ERR}}^{\text{MAP}})$. Combining with Lemma A.5 and Lemma A.6, we get

$$\begin{aligned} f_{\text{AUX}}(\psi) &\leq 3c \cdot \left(\mathbb{H}_2\left(p_{\text{ERR}}^{\text{MAP}}\right) + p_{\text{ERR}}^{\text{MAP}} \log m\right) + 4\left(\mathbb{H}_2(p_{\text{E}}) + p_{\text{E}} \log m\right) \\ &\leq (3c+4) \cdot \left(\mathbb{H}_2\left(p_{\text{ERR}}^{\text{MAP}}\right) + p_{\text{ERR}}^{\text{MAP}} \log m\right), \end{aligned}$$

which completes the proof. □

A.2.7 Proof of Theorem 4.19 Part 2: Proof of Lemma 4.21

In this section, we analyze the 1-step gain in the auxiliary function $\Delta_{\text{AUX}}(v \mid \psi)$, of any test $v \in \mathcal{V}$. By the end of this section, we will show that it is lowered bounded by the one-step gain in the EC^2 objective $\Delta_{\text{EC}^2,\psi}(v)$.

Recall that we assume test outcomes are binary for our analysis, and in the following of this section, we assume the outcome x_v of test v is in $\{+,-\}$ instead of $\{0,1\}$, for clarity purposes.

Notations and the Intermediate Goal

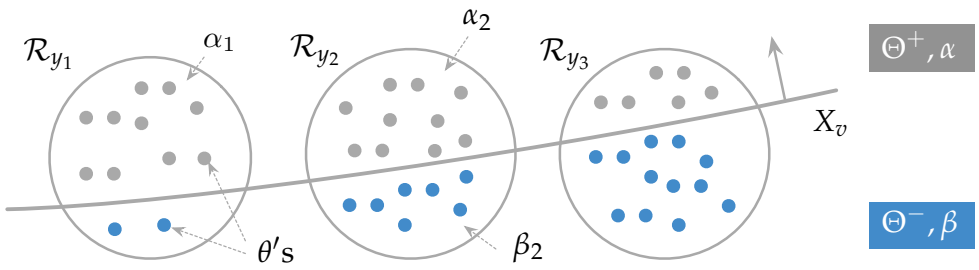


Figure A.6: Performing binary test v on Θ and Y. Dots represent root-causes $\theta \in \text{supp}(\Theta)$, and circles represent values of the target variable $y \in \mathcal{Y}$. The favorable outcome of X_v for the root-causes in solid dots are $+$; the favorable outcome for root-causes in hollow dots are $-$. We also illustrate the short-hand notations used in §A.2.7. They are: p, q (i.e., the posterior probability distribution over Y and Θ), γ (i.e., the prior distribution over Y and Θ) and α, β (i.e., the probability mass of solid and hollow dots, respectively, before performing test v).

Table A.2: Summary of notations introduced for the proof of Lemma 4.21.

γ	$\mathbb{P}[\cdot \mid \psi]$, i.e., probability distribution on Θ and Y, before performing test v
γ_+, γ_-	$\mathbb{P}[X_v = + \mid \psi]$,$\mathbb{P}[X_v = - \mid \psi]$
p_θ, p_y	$\mathbb{P}[\cdot \mid \psi, X_v = +]$, i.e., probability distribution on Θ and Y having observed $X_v = +$
q_θ, q_y	$\mathbb{P}[\cdot \mid \psi, X_v = -]$, i.e., probability distribution on Θ and Y having observed $X_v = -$
Θ^+, Θ^-	set of positive/ negative root-causes
Θ_i^+, Θ_i^-	set of positive/ negative root-causes associated with target y_i
α, β	total probability mass of positive/ negative root-causes
α_i, β_i	probability mass of positive/ negative root-causes associated with target y_i
μ_i, ν_i	α_i/α, β_i/β (defined in §A.2.7)
$\theta \nsim \theta'$	$r(\theta) \neq r(\theta')$, i.e., root-causes θ and θ' do not share the same target value

For brevity, we first define a few short-hand notations to simplify our derivation. Let p, q be two distributions on Θ, and $h = \gamma_+ p + \gamma_- q$ be the convex combination of the two, where $\gamma_+, \gamma_- \geq 0$ and $\gamma_+ + \gamma_- = 1$.

In fact, we are using p and q to refer to the posterior distribution over Θ *after* we observe the (noisy) outcome of some binary test v, and use γ to refer to the distribution over Θ *before* we perform the test, i.e., $p_\theta \triangleq \mathbb{P}[\theta \mid X_v = +]$, $q_\theta \triangleq \mathbb{P}[\theta \mid X_v = -]$, and $\gamma_\theta \triangleq \mathbb{P}[\theta] = \gamma_+ p_\theta + \gamma_- q_\theta$, where $\gamma_+ = \mathbb{P}[X_v = +]$ and $\gamma_- = \mathbb{P}[X_v = -]$. For $y_i \in \mathcal{Y}$, we use $p_i \triangleq \sum_{\theta : r(\theta) = y_i} p_\theta$ to denote the probability of y_i under distribution p, and use $q_i \triangleq \sum_{\theta : r(\theta) = y_i} q_\theta$ to denote the probability of y_i under distribution q.

Further, given a test v, we define Θ_i^+, Θ_i^- to be the set of root-causes associated with target y_i, whose favorable outcome of test v is $+$ (for Θ_i^+) and $-$ (for Θ_i^-). Formally,

$$\Theta_i^+ \triangleq \{\theta : r(\theta) = y_i \wedge \mathbb{P}[X_v = + \mid \theta] \geq 1/2\}$$
$$\Theta_i^- \triangleq \{\theta : r(\theta) = y_i \wedge \mathbb{P}[X_v = + \mid \theta] < 1/2\}$$

We then define $\Theta^+ \triangleq \bigcup_{i \in \{1,\ldots,n\}} \Theta_i^+$, and $\Theta^- \triangleq \bigcup_{i \in \{1,\ldots,n\}} \Theta_i^-$, to be the set of "positive" and "negative" root-causes for test v, respectively.

Let α_i, β_i be the probability mass of the root-causes in Θ_i^+ and Θ_i^-, i.e., $\alpha_i \triangleq \sum_{y\in\Theta_i^+} \mathbb{P}[\theta]$, and $\beta_i \triangleq \sum_{y\in\Theta_i^-} \mathbb{P}[\theta]$. We further define $\alpha \triangleq \sum_{y_i\in\mathcal{Y}} \alpha_i = \sum_{\theta\in\Theta^+} \mathbb{P}[\theta]$, and $\beta \triangleq \sum_{y_i\in\mathcal{Y}} \beta_y = \sum_{\theta\in\Theta^-} \mathbb{P}[\theta]$, then clearly we have $\alpha + \beta = 1$. See Fig. A.6 for illustration.

Now, we assume that test v has error rate ϵ. That is,

$$\forall\theta,\ \min\{\mathbb{P}[X_v = + \mid \theta], \mathbb{P}[X_v = - \mid \theta]\} = \epsilon.$$

Then, by definition of γ_+, γ_-, p_i, q_i, p_θ, q_θ, it is easy to verify that

$$\begin{aligned}
&\gamma_+ = \alpha\bar{\epsilon} + \beta\epsilon, \quad \gamma_- = \alpha\epsilon + \beta\bar{\epsilon} \\
&p_i = \frac{\alpha_i\bar{\epsilon} + \beta_i\epsilon}{\gamma_+}, \quad q_i = \frac{\alpha_i\epsilon + \beta_i\bar{\epsilon}}{\gamma_-} \\
&p_\theta = \frac{\gamma_\theta\bar{\epsilon}}{\gamma_+}, \quad q_\theta = \frac{\gamma_\theta\epsilon}{\gamma_-}, \qquad \text{if } \theta \in \Theta_i^+ \\
&p_\theta = \frac{\gamma_\theta\epsilon}{\gamma_+}, \quad q_\theta = \frac{\gamma_\theta\bar{\epsilon}}{\gamma_-}, \qquad \text{if } \theta \in \Theta_i^-
\end{aligned} \tag{A.2.20}$$

For the convenience of readers, we summarize the notations provided above in Table A.2.

Given root-causes θ and θ', we use $\theta \nsim \theta'$ to denote that the values of the target variable Y associated with root-causes θ and θ' are different, i.e., $r(\theta) \neq r(\theta')$.

We can rewrite the auxiliary function (as defined in Equation (4.5.3)) as follows:

$$f_{\text{AUX}} = \sum_{\theta\nsim\theta'} \gamma_\theta\gamma_{\theta'} \log\frac{1}{\gamma_\theta\gamma_{\theta'}} + c\sum_{y_i\in\mathcal{Y}} \mathbb{H}_2(\gamma_i).$$

If by performing test v we observe $X_v = +$, we have

$$f_{\text{AUX}}((v,+)) = \sum_{\theta\nsim\theta'} p_\theta p_{\theta'} \log\frac{1}{p_\theta p_{\theta'}} + c\sum_{y_i\in\mathcal{Y}} \mathbb{H}_2(p_i)$$

otherwise, if we observe $X_v = -$,

$$f_{\text{AUX}}((v,-)) = \sum_{\theta\nsim\theta'} q_\theta q_{\theta'} \log\frac{1}{q_\theta q_{\theta'}} + c\sum_{y_i\in\mathcal{Y}} \mathbb{H}_2(q_i)$$

Therefore, the expected gain (i.e., $\Delta_{\text{AUX}}(v \mid \psi)$) of performing test v is,

$$\Delta_{\text{AUX}} = \overbrace{\sum_{\theta\sim\theta'} \gamma_\theta\gamma_{\theta'} \log\frac{1}{\gamma_\theta\gamma_{\theta'}} - \left(\gamma_+ \sum_{\theta\sim\theta'} p_\theta p_{\theta'} \log\frac{1}{p_\theta p_{\theta'}} + \gamma_- \sum_{\theta\sim\theta'} q_\theta q_{\theta'} \log\frac{1}{q_\theta q_{\theta'}}\right)}^{\textcircled{1}} + \underbrace{c\left(\sum_{y_i\in\mathcal{Y}} \mathbb{H}_2(\gamma_i) - \left(\gamma_+ \sum_{y_i\in\mathcal{Y}} \mathbb{H}_2(p_i) + \gamma_- \sum_{y_i\in\mathcal{Y}} \mathbb{H}_2(q_i)\right)\right)}_{\textcircled{2}} \tag{A.2.21}$$

In the following, we derive lower bounds for the above two terms respectively.

A Lower Bound on Term 1

Let $g_{\theta,\theta'} \triangleq \gamma_+ p_\theta p_{\theta'} + \gamma_- q_\theta q_{\theta'}$. Then, we can rewrite Term ① as,

$$\text{Term } \textcircled{1} = \underbrace{\sum_{\theta\sim\theta'} \gamma_\theta\gamma_{\theta'} \log\frac{1}{\gamma_\theta\gamma_{\theta'}} - \sum_{\theta\sim\theta'} g_{\theta,\theta'} \log\frac{1}{g_{\theta,\theta'}}}_{\text{Part 1}} + \underbrace{\sum_{\theta\sim\theta'} g_{\theta,\theta'} \log\frac{1}{g_{\theta,\theta'}} - \left(\gamma_+ \sum_{\theta\sim\theta'} p_\theta p_{\theta'} \log\frac{1}{p_\theta p_{\theta'}} + \gamma_- \sum_{\theta\sim\theta'} q_\theta q_{\theta'} \log\frac{1}{q_\theta q_{\theta'}}\right)}_{\text{Part 2}} \tag{A.2.22}$$

Part 1. We first provide a lower bound for part 1 of Equation (A.2.22).

Notice that for concave function $f(x) = x\log\frac{1}{x}$ and $\delta < x$, it holds that $f(x) - f(x-\delta) \geq \delta\frac{\partial f}{\partial x}\big|_x = \delta(\log\frac{1}{x} - 1)$, then we get

$$\sum_{\theta\sim\theta'} \gamma_\theta\gamma_{\theta'} \log\frac{1}{\gamma_\theta\gamma_{\theta'}} - \sum_{\theta\sim\theta'} g_{\theta,\theta'} \log\frac{1}{g_{\theta,\theta'}} \geq \sum_{\theta\sim\theta'} (\gamma_\theta\gamma_{\theta'} - g_{\theta,\theta'}) \left(\log\frac{1}{\gamma_\theta\gamma_{\theta'}} - 1\right)$$

Further, observe

$$\begin{aligned}
\gamma_\theta\gamma_{\theta'} - g_{\theta,\theta'} &= (\gamma_+ p_\theta + \gamma_- q_\theta)(\gamma_+ p_{\theta'} + \gamma_- q_{\theta'}) - (\gamma_+ p_\theta p_{\theta'} + \gamma_- q_\theta q_{\theta'}) \\
&= (\gamma_+ p_\theta + \gamma_- q_\theta)(p_{\theta'} + q_{\theta'} - \gamma_- p_{\theta'} - \gamma_+ q_{\theta'}) - (\gamma_+ p_\theta p_{\theta'} + \gamma_- q_\theta q_{\theta'}) \\
&= \gamma_+\gamma_- p_{\theta'} q_\theta - \gamma_+\gamma_- p_{\theta'} p_\theta + \gamma_+\gamma_- p_\theta q_{\theta'} - \gamma_-\gamma_+ q_{\theta'} q_\theta \\
&= -\gamma_+\gamma_- (p_\theta - q_\theta)(p_{\theta'} - q_{\theta'})
\end{aligned}$$

Combining the above two equations gives us

$$\text{Part } 1 \geq \sum_{\theta \nsim \theta'} -\gamma_+\gamma_-(p_\theta - q_\theta)(p_{\theta'} - q_{\theta'}) \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right)$$

For any root-cause pair $\{\theta, \theta'\}$ with $\theta \nsim \theta'$, and binary test v, there are only 4 possible combinations regarding the root-causes' favorable outcomes. Namely,

1. Both θ and θ' maps x to $+$, i.e., $\theta \in \Theta^+ \wedge \theta' \in \Theta^+$.

   We define such set of root-cause pairs with positive favorable outcomes as $U_{(+,+)} \triangleq \{\{\theta, \theta'\} : \theta \in \Theta^+ \wedge \theta' \in \Theta^+\}$ (For other cases, we define $U_{(-,-)}$, $U_{(+,-)}$, $U_{(-,+)}$ in a similar way).

   In this case, we have

$$\begin{aligned}
&\sum_{\{\theta,\theta'\} \in U_{(+,+)}} -\gamma_+\gamma_-(p_\theta - q_\theta)(p_{\theta'} - q_{\theta'}) \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right) \\
\overset{\text{Eq (A.2.20)}}{=} &\sum_{\{\theta,\theta'\} \in U_{(+,+)}} -\gamma_+\gamma_- \left(\frac{\gamma_\theta \bar{\epsilon}}{\gamma_+} - \frac{\gamma_\theta \epsilon}{\gamma_-}\right)\left(\frac{\gamma_{\theta'} \bar{\epsilon}}{\gamma_+} - \frac{\gamma_{\theta'} \epsilon}{\gamma_-}\right) \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right) \\
= &\gamma_+\gamma_- \left(\frac{\gamma_- \bar{\epsilon} - \gamma_+ \bar{\epsilon}}{\gamma_+\gamma_-}\right)^2 \sum_{\{\theta,\theta'\} \in U_{(+,+)}} -\gamma_\theta \gamma_{\theta'} \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right) \\
= &\frac{\beta^2 (1-2\epsilon)^2}{\gamma_+\gamma_-} \sum_{\{\theta,\theta'\} \in U_{(+,+)}} -\gamma_\theta \gamma_{\theta'} \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right) \\
= &\frac{\beta^2 (1-2\epsilon)^2}{\gamma_+\gamma_-} \sum_{\{\theta,\theta'\} \in U_{(+,+)}} \left(-2\gamma_\theta \gamma_{\theta'} \log \frac{1}{\gamma_\theta} + \gamma_\theta \gamma_{\theta'}\right) \\
= &\frac{\beta^2 (1-2\epsilon)^2}{\gamma_+\gamma_-} \left(\sum_{y_i \in \mathcal{Y}} (\alpha - \alpha_i) \sum_{\theta \in \Theta_i^+} -2\gamma_\theta \log \frac{1}{\gamma_\theta} + \sum_{y_i \in \mathcal{Y}} \alpha_i (\alpha - \alpha_i)\right) \\
= &\frac{(1-2\epsilon)^2}{\gamma_+\gamma_-} \left(-2\beta^2 \sum_{y_i \in \mathcal{Y}} (\alpha - \alpha_i) \sum_{\theta \in \Theta_i^+} \gamma_\theta \log \frac{1}{\gamma_\theta} + \beta^2 \sum_{y_i \in \mathcal{Y}} \alpha_i (\alpha - \alpha_i)\right)
\end{aligned}$$

2. Both θ and θ' maps x to $-$. Similarly, we get

$$\begin{aligned}
&\sum_{\{\theta,\theta'\} \in U_{(-,-)}} -\gamma_+\gamma_-(p_\theta - q_\theta)(p_{\theta'} - q_{\theta'}) \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right) \\
= &\frac{(1-2\epsilon)^2}{\gamma_+\gamma_-} \left(-2\alpha^2 \sum_{y_i \in \mathcal{Y}} (\beta - \beta_i) \sum_{\theta \in \Theta_i^-} \gamma_\theta \log \frac{1}{\gamma_\theta} + \alpha^2 \sum_{y_i \in \mathcal{Y}} \beta_i (\beta - \beta_i)\right)
\end{aligned}$$

3. θ maps x to $+$, θ' maps x to $-$. We have

$$\begin{aligned}
&\sum_{(\theta,\theta')\in U_{(+,-)}} -\gamma_+\gamma_-(p_\theta - q_\theta)(p_{\theta'} - q_{\theta'}) \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right) \\
=&\frac{(1-2\epsilon)^2}{\gamma_+\gamma_-} \left(\alpha\beta \sum_{y_i\in\mathcal{Y}} (\beta - \beta_i) \sum_{\theta\in\Theta_i^+} \gamma_\theta \log \frac{1}{\gamma_\theta} + \alpha\beta \sum_{y_i\in\mathcal{Y}} (\alpha - \alpha_i) \sum_{\theta\in\Theta_i^-} \gamma_\theta \log \frac{1}{\gamma_\theta} \right. \\
&\qquad \left. -\alpha\beta \sum_{y_i\in\mathcal{Y}} \alpha_i(\beta - \beta_i) \right)
\end{aligned}$$

4. θ maps x to $-$, θ' maps x to $+$. By symmetry we have

$$\begin{aligned}
&\sum_{(\theta,\theta')\in U_{(-,+)}} -\gamma_+\gamma_-(p_\theta - q_\theta)(p_{\theta'} - q_{\theta'}) \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right) \\
=&\sum_{(\theta,\theta')\in U_{(+,-)}} -\gamma_+\gamma_-(p_\theta - q_\theta)(p_{\theta'} - q_{\theta'}) \left(\log \frac{1}{\gamma_\theta \gamma_{\theta'}} - 1\right)
\end{aligned}$$

Combining the above four equations, we obtain a lower bound on Part 1:

$$
\begin{aligned}
\text{Part 1} \geq \frac{(1-2\epsilon)^2}{\gamma_+\gamma_-} \Bigg(& -2\beta^2 \sum_{y_i \in \mathcal{Y}} (\alpha - \alpha_i) \sum_{\theta \in \Theta_i^+} \gamma_\theta \log \frac{1}{\gamma_\theta} + \beta^2 \sum_{y_i \in \mathcal{Y}} \alpha_i (\alpha - \alpha_i) \\
& -2\alpha^2 \sum_{y_i \in \mathcal{Y}} (\beta - \beta_i) \sum_{\theta \in \Theta_i^-} \gamma_\theta \log \frac{1}{\gamma_\theta} + \alpha^2 \sum_{y_i \in \mathcal{Y}} \beta_i (\beta - \beta_i) \\
& +2\alpha\beta \sum_{y_i \in \mathcal{Y}} (\beta - \beta_i) \sum_{\theta \in \Theta_i^+} \gamma_\theta \log \frac{1}{\gamma_\theta} \\
& +2\alpha\beta \sum_{y_i \in \mathcal{Y}} (\alpha - \alpha_i) \sum_{\theta \in \Theta_i^-} \gamma_\theta \log \frac{1}{\gamma_\theta} - 2\alpha\beta \sum_{y_i \in \mathcal{Y}} \alpha_i (\beta - \beta_i) \Bigg) \\
= \frac{(1-2\epsilon)^2}{\gamma_+\gamma_-} \Bigg(& \left(2\alpha\beta \sum_{y_i \in \mathcal{Y}} (\beta - \beta_i) - 2\beta^2 \sum_{y_i \in \mathcal{Y}} (\alpha - \alpha_i) \right) \sum_{\theta \in \Theta_i^+} \gamma_\theta \log \frac{1}{\gamma_\theta} \\
& + \left(2\alpha\beta \sum_{y_i \in \mathcal{Y}} (\alpha - \alpha_i) - 2\alpha^2 \sum_{y_i \in \mathcal{Y}} (\beta - \beta_i) \right) \sum_{\theta \in \Theta_i^-} \gamma_\theta \log \frac{1}{\gamma_\theta} \\
& +\beta^2 \sum_{y_i \in \mathcal{Y}} \alpha_i (\alpha - \alpha_i) + \alpha^2 \sum_{y_i \in \mathcal{Y}} \beta_i (\beta - \beta_i) - 2\alpha\beta \sum_{y_i \in \mathcal{Y}} \alpha_i (\beta - \beta_i) \Bigg) \\
= \frac{(1-2\epsilon)^2}{\gamma_+\gamma_-} \cdot \Bigg(& 2 \sum_{y_i \in \mathcal{Y}} \beta(\beta\alpha_i - \alpha\beta_i) \sum_{\theta \in \Theta_i^+} \gamma_\theta \log \frac{1}{\gamma_\theta} \\
& +2 \sum_{y_i \in \mathcal{Y}} \alpha(\alpha\beta_i - \beta\alpha_i) \sum_{\theta \in \Theta_i^-} \gamma_\theta \log \frac{1}{\gamma_\theta} - \sum_{y_i \in \mathcal{Y}} (\beta\alpha_i - \alpha\beta_i)^2 \Bigg) \\
= \frac{(1-2\epsilon)^2}{\gamma_+\gamma_-} \cdot \Bigg(& 2 \sum_{y_i \in \mathcal{Y}} (\beta\alpha_i - \alpha\beta_i) \left(\beta\alpha_i \sum_{\theta \in \Theta_i^+} \frac{\gamma_\theta}{\alpha_i} \log \frac{1}{\gamma_\theta} - \alpha\beta_i \sum_{\theta \in \Theta_i^-} \frac{\gamma_\theta}{\beta_i} \log \frac{1}{\gamma_\theta} \right) - \\
& \sum_{y_i \in \mathcal{Y}} (\beta\alpha_i - \alpha\beta_i)^2 \Bigg)
\end{aligned}
\tag{A.2.23}
$$

Part 2. Next, we will provide a lower bound on Part 2 of Equation (A.2.22).

By definition, we have

$$\begin{aligned}\text{Part 2} = &\sum_{\theta\sim\theta'} (\gamma_+ p_\theta p_{\theta'} + \gamma_- q_\theta q_{\theta'}) \log \frac{1}{\gamma_+ p_\theta p_{\theta'} + \gamma_- q_\theta q_{\theta'}} \\ &- \left(\gamma_+ \sum_{\theta\sim\theta'} p_\theta p_{\theta'} \log \frac{1}{p_\theta p_{\theta'}} + \gamma_- \sum_{\theta\sim\theta'} q_\theta q_{\theta'} \log \frac{1}{q_\theta q_{\theta'}} \right) \\ \overset{(a)}{\geq} &\frac{\gamma_+\gamma_-}{2} \sum_{\theta\sim\theta'} \frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}}\end{aligned}$$

Hereby, step (a) is due to the strong concavity[1] of $f(x) = x \log \frac{1}{x}$.

As with the analysis of Part 1, we consider the four sets of $\{\theta, \theta'\}$ pairs:

1. $\{\theta, \theta'\} \in U_{(+,+)}$: both θ and θ' maps x to $+$.

   In this case, we have

$$\begin{aligned}\sum_{(\theta,\theta')\in U_{(+,+)}} \frac{\gamma_+\gamma_-}{2} \frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}} &\geq \sum_{(\theta,\theta')\in U_{(+,+)}} \frac{\gamma_+\gamma_-}{2} (\sqrt{p_\theta p_{\theta'}} - \sqrt{q_\theta q_{\theta'}})^2 \\ &\overset{\text{Eq (A.2.20)}}{=} \sum_{(\theta,\theta')\in U_{(+,+)}} \frac{\gamma_+\gamma_-}{2} \left(\sqrt{\frac{\gamma_\theta \bar{\epsilon}}{\gamma_+} \frac{\gamma_{\theta'} \bar{\epsilon}}{\gamma_+}} - \sqrt{\frac{\gamma_\theta \epsilon}{\gamma_-} \frac{\gamma_{\theta'} \epsilon}{\gamma_-}} \right)^2 \\ &= \sum_{(\theta,\theta')\in U_{(+,+)}} \frac{\gamma_+\gamma_-}{2} \gamma_\theta \gamma_{\theta'} \left(\frac{\bar{\epsilon}}{\gamma_+} - \frac{\epsilon}{\gamma_-} \right)^2 \\ &= \sum_{(\theta,\theta')\in U_{(+,+)}} \frac{\gamma_+\gamma_-}{2} \gamma_\theta \gamma_{\theta'} \frac{\beta^2 (1-2\epsilon)^2}{(\gamma_+\gamma_-)^2} \\ &= \frac{(1-2\epsilon)^2}{2\gamma_+\gamma_-} \beta^2 \sum_{y_i\in\mathcal{Y}} \alpha_i(\alpha - \alpha_i)\end{aligned}$$

2. $(\theta, \theta') \in U_{(-,-)}$. Similarly, we get

$$\sum_{(\theta,\theta')\in U_{(-,-)}} \frac{\gamma_+\gamma_-}{2} \frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}} \geq \frac{(1-2\epsilon)^2}{2\gamma_+\gamma_-} \alpha^2 \sum_{y_i\in\mathcal{Y}} \beta_i(\beta - \beta_i)$$

[1] If f is strongly concave, then for $t \in [0,1]$, it holds that $f(tx + (1-t)y) - tf(x) - (1-t)f(y) \geq \frac{t(1-t)}{2} m(x-y)^2$, where $m = \min(|f''(x)|, |f''(y)|)$.

3. $(\theta, \theta') \in U_{(+,-)}$: θ maps x to $+$, θ' maps x to $-$. We have

$$\begin{aligned}\sum_{(\theta,\theta')\in U_{(+,-)}} \frac{\gamma_+\gamma_-}{2}\frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}} &\geq \sum_{(\theta,\theta')\in U_{(+,+)}} \frac{\gamma_+\gamma_-}{2}\left(\sqrt{\frac{\gamma_\theta\bar{\epsilon}}{\gamma_+}\frac{\gamma_{\theta'}\epsilon}{\gamma_+}} - \sqrt{\frac{\gamma_\theta\epsilon}{\gamma_-}\frac{\gamma_{\theta'}\bar{\epsilon}}{\gamma_-}}\right)^2 \\ &= \sum_{(\theta,\theta')\in U_{(+,+)}} \frac{\gamma_+\gamma_-}{2}\gamma_\theta\gamma_{\theta'}\epsilon\bar{\epsilon}\left(\frac{1}{\gamma_+} - \frac{1}{\gamma_-}\right)^2 \\ &= \frac{(1-2\epsilon)^2}{2\gamma_+\gamma_-}\epsilon\bar{\epsilon}(\alpha-\beta)^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\end{aligned}$$

4. $(\theta, \theta') \in U_{(-,+)}$: θ maps x to $-$, θ' maps x to $+$. By symmetry we have

$$\sum_{(\theta,\theta')\in U_{(+,-)}} \frac{\gamma_+\gamma_-}{2}\frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}} \geq \frac{(1-2\epsilon)^2}{2\gamma_+\gamma_-}\epsilon\bar{\epsilon}(\alpha-\beta)^2\sum_{y_i\in\mathcal{Y}}\beta_i(\alpha-\alpha_i)$$

Combining the above four equations, we obtain a lower bound on Part 2:

$$\begin{aligned}\text{Part 2} \geq &\sum_{(\theta,\theta')\in U_{(+,+)}} \frac{\gamma_+\gamma_-}{2}\frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}} + \sum_{(\theta,\theta')\in U_{(-,-)}} \frac{\gamma_+\gamma_-}{2}\frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}} \\ &+ \sum_{(\theta,\theta')\in U_{(+,-)}} \frac{\gamma_+\gamma_-}{2}\frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}} + \sum_{(\theta,\theta')\in U_{(-,+)}} \frac{\gamma_+\gamma_-}{2}\frac{(p_\theta p_{\theta'} - q_\theta q_{\theta'})^2}{p_\theta p_{\theta'} + q_\theta q_{\theta'}} \\ = &\frac{(1-2\epsilon)^2}{2\gamma_+\gamma_-}\left(\beta^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i) + \alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i) + 2\epsilon\bar{\epsilon}(\alpha-\beta)^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\right)\end{aligned} \tag{A.2.24}$$

A Lower Bound on Term 2

Now we move on to analyze Term ❷ of Equation (A.2.22). By strong concavity of $f(x) = x\log\frac{1}{x} + (1-x)\log\frac{1}{1-x}$, we obtain

$$\begin{aligned}\text{Term } ❷ &= c\sum_{y_i\in\mathcal{Y}}\left(\gamma_i\log\frac{1}{\gamma_i} + (1-\gamma_i)\log\frac{1}{1-\gamma_i} - \gamma_+\left(p_i\log\frac{1}{p_i} + (1-p_i)\log\frac{1}{1-p_i}\right)\right. \\ &\qquad \left. - \gamma_-\left(q_i\log\frac{1}{q_i} + (1-q_i)\log\frac{1}{1-q_i}\right)\right) \\ &\overset{\text{footnote 1}}{\geq} \frac{c\cdot\gamma_+\gamma_-}{2}\sum_{y_i\in\mathcal{Y}}\frac{(p_i-q_i)^2}{\max\{p_i(1-p_i), q_i(1-q_i)\}}\end{aligned}$$

Plugging in the definition of p_i, q_i from Equation (A.2.20), we get

$$\begin{aligned}
\text{Term } ② &= \frac{c \cdot \gamma_+\gamma_-}{2} \sum_{y_i \in \mathcal{Y}} \left(\frac{\alpha_i \bar{\epsilon} + \beta_i \epsilon}{\gamma_+} - \frac{\alpha_i \epsilon + \beta_i \bar{\epsilon}}{\gamma_-} \right)^2 \frac{1}{\max\{p_i(1-p_i), q_i(1-q_i)\}} \\
&= \frac{c}{2\gamma_+\gamma_-} \sum_{y_i \in \mathcal{Y}} \frac{((\alpha\epsilon + \beta\bar{\epsilon})(\alpha_i\bar{\epsilon} + \beta_i\epsilon) - (\alpha\bar{\epsilon} + \beta\epsilon)(\alpha_i\epsilon + \beta_i\bar{\epsilon}))^2}{\max\{p_i(1-p_i), q_i(1-q_i)\}} \\
&= \frac{c}{2\gamma_+\gamma_-} \sum_{y_i \in \mathcal{Y}} \frac{\left(\alpha\beta_i\epsilon^2 + \beta\alpha_i\bar{\epsilon}^2 - \alpha\beta_i\bar{\epsilon}^2 - \beta\alpha_i\epsilon^2\right)^2}{\max\{p_i(1-p_i), q_i(1-q_i)\}} \\
&= \frac{c(1-2\epsilon)^2}{2\gamma_+\gamma_-} \sum_{y_i \in \mathcal{Y}} \frac{(\beta\alpha_i - \alpha\beta_i)^2}{\max\{p_i(1-p_i), q_i(1-q_i)\}}
\end{aligned} \tag{A.2.25}$$

A Combined Lower Bound for Δ_{aux}

Now, combining Equation (A.2.23), (A.2.24), and (A.2.25), we can get a lower bound for Δ_{AUX}:

$$\begin{aligned}
\Delta_{\text{AUX}} \geq \frac{(1-2\epsilon)^2}{\gamma_+\gamma_-} \cdot \Bigg(& 2 \sum_{y_i \in \mathcal{Y}} (\beta\alpha_i - \alpha\beta_i) \left(\beta\alpha_i \sum_{\theta \in \Theta_i^+} \frac{\gamma_\theta}{\alpha_i} \log \frac{1}{\gamma_\theta} - \alpha\beta_i \sum_{\theta \in \Theta_i^-} \frac{\gamma_\theta}{\beta_i} \log \frac{1}{\gamma_\theta} \right) \\
& - \sum_{y_i \in \mathcal{Y}} (\beta\alpha_i - \alpha\beta_i)^2 \Bigg) \\
& + \frac{(1-2\epsilon)^2}{2\gamma_+\gamma_-} \left(\beta^2 \sum_{y_i \in \mathcal{Y}} \alpha_i(\alpha - \alpha_i) + \alpha^2 \sum_{y_i \in \mathcal{Y}} \beta_i(\beta - \beta_i) + 2\epsilon\bar{\epsilon}(\alpha - \beta)^2 \sum_{y_i \in \mathcal{Y}} \alpha_i(\beta - \beta_i) \right) \\
& + \frac{c(1-2\epsilon)^2}{2\gamma_+\gamma_-} \sum_{y_i \in \mathcal{Y}} \frac{(\beta\alpha_i - \alpha\beta_i)^2}{\max\{p_i(1-p_i), q_i(1-q_i)\}}
\end{aligned} \tag{A.2.26}$$

We can rewrite Equation (A.2.26) as

$$\Delta_{\text{AUX}} \geq \underbrace{\frac{(1-2\epsilon)^2}{4\gamma_+\gamma_-}\left(\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)^2+\beta^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+2\epsilon\bar{\epsilon}(\alpha-\beta)^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\right)}_{\text{LB1}}$$
$$+\underbrace{\frac{(1-2\epsilon)^2}{4\gamma_+\gamma_-}\left(\beta^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+2\epsilon\bar{\epsilon}(\alpha-\beta)^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)+2c\sum_{y_i\in\mathcal{Y}}\frac{(\beta\alpha_i-\alpha\beta_i)^2}{\max\{p_i(1-p_i),q_i(1-q_i)\}}-5\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)^2+8\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)\left(\beta\alpha_i\sum_{\theta\in\Theta_i^+}\frac{\gamma_\theta}{\alpha_i}\log\frac{1}{\gamma_\theta}-\alpha\beta_i\sum_{\theta\in\Theta_i^-}\frac{\gamma_\theta}{\beta_i}\log\frac{1}{\gamma_\theta}\right)\right)}_{\text{LB2}} \tag{A.2.27}$$

Connecting Δ_{aux} with Δ_{EC^2}

Next, we will show that term LB1 is lower-bounded by a factor of Δ_{EC^2} (i.e., $\Delta_{\text{EC}^2,\psi}(v)$), while LB2 cannot be too much less than 0. Concretely, we will show

- LB1 $\geq \frac{1}{16}(1-2\epsilon)^2\Delta_{\text{EC}^2}$, and
- LB2 $\geq -2n(1-2\epsilon)^2\eta$, for $\eta\in(0,1)$.

At the end of this subsection, we will combine the above results to connect $\Delta_{\text{AUX}}(v \mid \psi)$ with $\Delta_{\text{EC}^2,\psi}(v)$ (See Equation (A.2.34)).

LB1 VS. Δ_{EC^2}. We expand the EC^2 gain $\Delta_{\text{EC}^2,\psi}(v)$ as

$$\begin{aligned}\Delta_{\text{EC}^2} &= \sum_{y_i\in\mathcal{Y}}(\alpha_i+\beta_i)(1-\alpha_i-\beta_i)-\alpha\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)-\beta\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)\\ &= \beta\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\end{aligned} \tag{A.2.28}$$

Define

$$\begin{cases} \textcircled{*} & \triangleq \frac{16\gamma_+\gamma_-}{(1-2\epsilon)^2} \cdot \mathrm{LB1} \\ & = 4\left(\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)^2+\beta^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)\right. \\ & \qquad\qquad \left.+2\epsilon\bar{\epsilon}(\alpha-\beta)^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\right) \\ \textcircled{\#} & \triangleq \gamma_+\gamma_-\Delta_{\mathrm{EC}^2} \\ & = \left(\epsilon\bar{\epsilon}(\alpha-\beta)^2+\alpha\beta\right)\left(\beta\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\right) \end{cases}$$

To bound LB1 against $\frac{1}{16}(1-2\epsilon)^2\Delta_{\mathrm{EC}^2}$, it suffices to show $\textcircled{*} \geq \textcircled{\#}$.

To prove the above inequality, we consider the following two cases:

1. $\epsilon\bar{\epsilon}(\alpha-\beta)^2 \leq \alpha\beta$. In this case, we have $\epsilon\bar{\epsilon}(\alpha-\beta)^2+\alpha\beta \leq 2\alpha\beta$. Then,

$$\begin{aligned} \frac{\textcircled{*}-\textcircled{\#}}{2} &\geq \frac{\textcircled{*}}{2}-\alpha\beta\left(\beta\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\right) \\ &\geq \beta^2(1+\beta)\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha^2(1+\alpha)\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)-2\alpha\beta\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i) \\ &\qquad\qquad +\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)^2 \\ &\geq \beta^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)-2\alpha\beta\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)+\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)^2 \\ &= 0 \end{aligned}$$

2. $\epsilon\bar{\epsilon}(\alpha-\beta)^2 > \alpha\beta$. W.l.o.g., we assume $\beta \leq \alpha \leq 1$. By $\alpha+\beta=1$ we get $2\alpha \geq 1$.

   Observe the fact that

$$\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)^2 = -\beta^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)-\alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+2\alpha\beta\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i) \geq 0$$

   Rearranging the terms in the above inequality, we get

$$\beta\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i) \leq 2\alpha\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i) \leq 2(\alpha\beta-\sum_{y_i\in\mathcal{Y}}\alpha_i\beta_i) = 2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i) \tag{A.2.29}$$

Hence,

$$
\begin{aligned}
\text{\#} &\leq 2\epsilon\bar{\epsilon}(\alpha-\beta)^2\left(\beta\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\right)\\
&\overset{(A.2.29)}{\leq} 2\epsilon\bar{\epsilon}(\alpha-\beta)^2\left(\alpha\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+4\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\right)\\
&\overset{2\alpha\geq 1}{\leq} 2\epsilon\bar{\epsilon}(\alpha-\beta)^2\left(2\alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)+4\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)\right)\\
&\overset{\epsilon\bar{\epsilon}(\alpha-\beta)^2\leq 1}{\leq} 4\left(2\epsilon\bar{\epsilon}(\alpha-\beta)^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\beta-\beta_i)+\alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)\right)\\
&\leq \text{*}
\end{aligned}
$$

Therefore, we get

$$\mathrm{LB1}\geq\frac{1}{16}(1-2\epsilon)^2\,\Delta_{\mathrm{EC}^2} \tag{A.2.30}$$

A lower bound on LB2. In the following, we will analyze LB2.

$$
\begin{aligned}
\mathrm{LB2}\geq\frac{(1-2\epsilon)^2}{4\gamma_+\gamma_-}&\left(\beta^2\sum_{y_i\in\mathcal{Y}}\alpha_i(\alpha-\alpha_i)+\alpha^2\sum_{y_i\in\mathcal{Y}}\beta_i(\beta-\beta_i)-5\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)^2\right.\\
&+2c_2\sum_{y_i\in\mathcal{Y}}\frac{(\beta\alpha_i-\alpha\beta_i)^2}{\max\{p_i(1-p_i),q_i(1-q_i)\}}\\
&\left.+8\sum_{y_i\in\mathcal{Y}}(\beta\alpha_i-\alpha\beta_i)\left(\beta\alpha_i\sum_{\theta\in\Theta_i^+}\frac{\gamma_\theta}{\alpha_i}\log\frac{\alpha_i}{\gamma_\theta}+\beta\alpha_i\log\frac{1}{\alpha_i}-\alpha\beta_i\sum_{\theta\in\Theta_i^-}\frac{\gamma_\theta}{\beta_i}\log\frac{\beta_i}{\gamma_\theta}-\alpha\beta_i\log\frac{1}{\beta_i}\right)\right.
\end{aligned}
$$

For brevity, define $\mu_i\triangleq\alpha_i/\alpha$, and $\nu_i\triangleq\beta_i/\beta$. We can simplify the above equation as

$$
\begin{aligned}
\mathrm{LB2}\geq\frac{\alpha^2\beta^2(1-2\epsilon)^2}{4\gamma_+\gamma_-}\sum_{y_i\in\mathcal{Y}}&\left(\mu_i(1-\mu_i)+\nu_i(1-\nu_i)-5(\mu_i-\nu_i)^2+\frac{2c_2(\mu_i-\nu_i)^2}{\max\{p_i(1-p_i),q_i(1-q_i)\}}\right.\\
&\left.+8(\mu_i-\nu_i)\left(\mu_i\sum_{\theta\in\Theta_i^+}\frac{\gamma_\theta}{\alpha_i}\log\frac{\alpha_i}{\gamma_\theta}+\mu_i\log\frac{1}{\mu_i\alpha}-\nu_i\sum_{\theta\in\Theta_i^-}\frac{\gamma_\theta}{\beta_i}\log\frac{\beta_i}{\gamma_\theta}-\nu_i\log\frac{1}{\nu_i\beta}\right)\right)
\end{aligned}
\tag{A.2.31}
$$

Denote the summand on the RHS of the above equation as $\mathrm{LB2}_i$. If for any $y_i\in\mathcal{Y}$ we can lower bound $\mathrm{LB2}_i$, we can then bound the whole sum. Fix i. W.l.o.g., we assume

$\mu_i \geq \nu_i$. Then

$$\begin{aligned}
\text{LB2}_i \triangleq\ & \mu_i(1-\mu_i) + \nu_i(1-\nu_i) - 5(\mu_i-\nu_i)^2 + \frac{2c\,(\mu_i-\nu_i)^2}{\max\{p_i(1-p_i), q_i(1-q_i)\}} \\
& + 8(\mu_i-\nu_i)\left(\mu_i \sum_{\theta\in\Theta_i^+} \frac{\gamma_\theta}{\alpha_i}\log\frac{\alpha_i}{\gamma_\theta} + \mu_i \log\frac{1}{\mu_i\alpha} \geq 0 \quad - \nu_i \sum_{\theta\in\Theta_i^-} \frac{\gamma_\theta}{\beta_i}\log\frac{\beta_i}{\gamma_\theta} - \nu_i\log\frac{1}{\nu_i\beta}\right) \\
\geq\ & \mu_i(1-\mu_i) + \nu_i(1-\nu_i) - 5(\mu_i-\nu_i)^2 + \frac{2c\,(\mu_i-\nu_i)^2}{\max\{p_i(1-p_i), q_i(1-q_i)\}} \\
& - 8(\mu_i-\nu_i)\left(\nu_i \sum_{\theta\in\Theta_i^-} \frac{\gamma_\theta}{\beta_i}\log\frac{\beta_i}{\gamma_\theta} \leq \log m \quad + \nu_i\log\frac{1}{\nu_i} + \nu_i\log\frac{1}{\beta}\right) \\
\geq\ & \mu_i(1-\mu_i) + \nu_i(1-\nu_i) - 5(\mu_i-\nu_i)^2 - 8(\mu_i-\nu_i)\left(\nu_i\log\frac{m}{\beta} + \nu_i\log\frac{1}{\nu_i}\right) \\
& + \frac{2c\,(\mu_i-\nu_i)^2}{\max\{p_i(1-p_i), q_i(1-q_i)\}}
\end{aligned}$$

To put a lower bound on the above terms, we first need to lower bound the term involving $\frac{(\mu_i-\nu_i)^2}{\max\{p_i(1-p_i),q_i(1-q_i)\}}$. Notice that $p_i = \frac{\alpha_i+\beta_i\epsilon/\bar{\epsilon}}{\alpha+\beta\epsilon/\bar{\epsilon}}$, and $p_i = \frac{\alpha_i\epsilon/\bar{\epsilon}+\beta_i}{\alpha\epsilon/\bar{\epsilon}+\beta}$. Therefore, $\min\{\mu_i,\nu_i\} \leq p_i, q_i \leq \max\{\mu_i,\nu_i\}$.

We check three different cases:

- $\mu_i \geq \nu_i \geq 1/2$, or $\nu_i \leq \mu_i \leq 1/2$.

In this case, $\max\{p_i(1-p_i), q_i(1-q_i)\} \le \max\{\mu_i(1-\mu_i), \nu_i(1-\nu_i)\}$. Therefore,

$$\begin{aligned}
\mathrm{LB2}_i &\ge -5(\mu_i-\nu_i)^2 - 8(\mu_i-\nu_i)\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right) \\
&\quad + \frac{2c\,(\mu_i-\nu_i)^2}{\max\{\mu_i(1-\mu_i), \nu_i(1-\nu_i)\}} + \mu_i(1-\mu_i) + \nu_i(1-\nu_i) \\
&\ge -5(\mu_i-\nu_i)^2 - 8(\mu_i-\nu_i)\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right) \\
&\quad + \frac{2c\,(\mu_i-\nu_i)^2}{\max\{\mu_i(1-\mu_i), \nu_i(1-\nu_i)\}} + \max\{\mu_i(1-\mu_i), \nu_i(1-\nu_i)\} \\
&\ge -5(\mu_i-\nu_i)^2 - 8(\mu_i-\nu_i)\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right) + 2\sqrt{2c}(\mu_i-\nu_i) \\
&\overset{\mu_i-\nu_i\le 1/2}{\ge} (\mu_i-\nu_i)\left(2\sqrt{2c} - 5/2 - 8\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right)\right) \\
&\overset{\text{(a)}}{\ge} (\mu_i-\nu_i)\left(2\sqrt{2c} - 5/2 - 8\log\frac{m}{\beta}\right)
\end{aligned}$$

Here, step (a) is because $f(x) = x\log\frac{m}{\beta x}$ is monotone increasing for $n \ge 3$. When $n < 3$, we have $\mu_i = 1$ and $\nu_i = 0$ (otherwise, there is no uncertainty left in Y) and hence the problem becomes trivial.

- $1/n \le \nu_i \le 1/2 \le \mu_i$.

  In this case, we cannot replace p_i, q_i with μ_i or ν_i. However, notice that $\max\{\mu_i(1-\mu_i), \nu_i(1-\nu_i)\} \le 1/4$, we have

$$\begin{aligned}
\mathrm{LB2}_i &\ge \mu_i(1-\mu_i) + \nu_i(1-\nu_i) - 5(\mu_i-\nu_i)^2 - 8(\mu_i-\nu_i)\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right) \\
&\quad + 8c\,(\mu_i-\nu_i)^2 \\
&= \mu_i(1-\mu_i) + \nu_i(1-\nu_i) + (\mu_i-\nu_i)^2 + (8c-6)(\mu_i-\nu_i)^2 \\
&\quad - 8(\mu_i-\nu_i)\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right) \\
&= \mu_i(1-\nu_i) + \nu_i(1-\mu_i) + (8c-6)(\mu_i-\nu_i)^2 - 8(\mu_i-\nu_i)\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right) \\
&\ge \mu_i(1-\nu_i) + (8c-6)(\mu_i-\nu_i)^2 - 8(\mu_i-\nu_i)\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right) \qquad \text{(A.2.32)} \\
&\overset{\nu_i\ge 1/n}{\ge} \mu_i(1-\nu_i) + (8c-6)(\mu_i-\nu_i)^2 - 8(\mu_i-\nu_i)\nu_i \log\frac{m^2}{\beta}
\end{aligned}$$

To further simplify notation, we denote $\gamma_1 \triangleq 8c - 6$, and $\gamma_2 \triangleq 8 \log \frac{m^2}{\beta}$. Then the above equation can be rewritten as

$$\mathrm{LB2}_i \geq \mu_i(1-\nu_i) + \gamma_1(\mu_i - \nu_i)^2 - \gamma_2(\mu_i - \nu_i)\nu_i$$

If $\mu_i - \nu_i \leq \frac{1}{2\gamma_2}$, then

$$\mathrm{LB2}_i \geq \mu_i(1-\nu_i) + \gamma_1(\mu_i - \nu_i)^2 - \frac{1}{2\gamma_2}\gamma_2\nu_i = \mu_i(1-\nu_i) - \frac{\nu_i}{2} \geq 0$$

Otherwise, if $\mu_i - \nu_i > \frac{1}{2\gamma_2}$, we have

$$\begin{aligned}
\mathrm{LB2}_i &\geq \mu_i(1-\nu_i) + (\mu_i - \nu_i)\left(\gamma_1(\mu_i - \nu_i) - \gamma_2\nu_i\right) \\
&> \mu_i(1-\nu_i) + (\mu_i - \nu_i)\left(\gamma_1\frac{1}{2\gamma_2} - \gamma_2\nu_i\right) \\
&> \frac{\mu_i - \nu_i}{2}\left(\frac{\gamma_1}{\gamma_2} - \gamma_2\right)
\end{aligned}$$

- $\nu_i \leq 1/n < 1/2 \leq \mu_i$. In this case, we have

$$\begin{aligned}
\mathrm{LB2}_i &\overset{\text{Eq (A.2.32)}}{\geq} \mu_i(1-\nu_i) + \gamma_1(\mu_i - \nu_i)^2 - 8(\mu_i - \nu_i)\left(\nu_i \log\frac{m}{\beta} + \nu_i \log\frac{1}{\nu_i}\right) \\
&\geq \mu_i(1-\nu_i) + \gamma_1(\mu_i - \nu_i)^2 - 8(\mu_i - \nu_i)\left(\frac{1}{m}\log\frac{m}{\beta} + \frac{\log m}{m}\right) \\
&= \mu_i(1-\nu_i) + \gamma_1(\mu_i - \nu_i)^2 - \frac{\gamma_2}{m}(\mu_i - \nu_i) \\
&> \mu_i(1-\nu_i) + (\mu_i - \nu_i)\left(\gamma_1\frac{n-2}{2n} - \frac{\gamma_2}{m}\right) \\
&\overset{(a)}{\geq} \frac{\mu_i - \nu_i}{3}\left(\frac{\gamma_1}{2} - \gamma_2\right) \\
&\geq \frac{\mu_i - \nu_i}{3}\left(\frac{\gamma_1}{\gamma_2} - \gamma_2\right)
\end{aligned}$$

  Step (a) is because $1/n < 1/2$ and therefore $n \geq 3$.

Putting the above cases together, we obtain the following equations:

$$\mathrm{LB2}_i \geq \begin{cases}
(\mu_i - \nu_i)\left(2\sqrt{2c} - 5/2 - 8\log\frac{m}{\beta}\right) & \text{if } \mu_i \geq \nu_i \geq 1/2, \text{ or } \nu_i \leq \mu_i \leq 1/2 \\
0 & \text{if } 1/n \leq \nu_i \leq 1/2 \leq \mu_i, \text{ and } \mu_i - \nu_i \leq \frac{1}{2\gamma_2} \\
\frac{\mu_i - \nu_i}{2}\left(\frac{\gamma_1}{\gamma_2} - \gamma_2\right) & \text{if } 1/n \leq \nu_i \leq 1/2 \leq \mu_i, \text{ and } \mu_i - \nu_i > \frac{1}{2\gamma_2} \\
\frac{\mu_i - \nu_i}{3}\left(\frac{\gamma_1}{\gamma_2} - \gamma_2\right) & \text{if } \nu_i \leq 1/n < 1/2 \leq \mu_i
\end{cases}$$

Fix $\eta \geq 0$. Let $c = 8\left(\log \frac{2m^2}{\eta}\right)^2$, we have $\gamma_1 > \left(8 \log \frac{m^2}{\eta}\right)^2$, and $\gamma_2 = 8 \log \frac{m^2}{\beta}$, so

$$\frac{\gamma_1}{\gamma_2} - \gamma_2 = \frac{(\sqrt{\gamma_1} - \gamma_2)(\sqrt{\gamma_1} + \gamma_2)}{\gamma_2} > 8\frac{\sqrt{\gamma_1} + \gamma_2}{\gamma_2} \log \frac{\beta}{\eta}$$

and thus we get

$$\mathrm{LB2}_i \geq \begin{cases} 8(\mu_i - \nu_i) \log \frac{\beta}{\eta} & \text{if } \mu_i \geq \nu_i \geq 1/2, \text{ or } \nu_i \leq \mu_i \leq 1/2 \\ 0 & \text{if } 1/n \leq \nu_i \leq 1/2 \leq \mu_i, \text{ and } \mu_i - \nu_i \leq \frac{1}{2\gamma_2} \\ \frac{4(\mu_i - \nu_i)(\sqrt{\gamma_1} + \gamma_2)}{\gamma_2} \log \frac{\beta}{\eta} & \text{if } \nu_i \leq 1/2 \leq \mu_i, \text{ and } \mu_i - \nu_i > \frac{1}{2\gamma_2} \end{cases}$$

That is, if $\beta \geq \eta$, we have $\mathrm{LB2}_i \geq 0$ for all $i \in \{1, \ldots, n\}$.

On the other hand, if $\beta < \eta$, we get $\frac{4(\sqrt{\gamma_1} + \gamma_2)}{\gamma_2} = \frac{4(\log \frac{m^2}{\eta} + \log \frac{m^2}{\beta})}{\log \frac{m^2}{\beta}} \leq 8$, and therefore $\mathrm{LB2}_i \geq 8(\mu_i - \nu_i) \log \frac{\beta}{\eta}$.

Summing over all $i \in \{1, \ldots, n\}$, we get that for $\beta < \eta$, it holds $\mathrm{LB2} \geq \sum_{y_i \in \mathcal{Y}} |\mu_i - \nu_i| \cdot \frac{2\alpha^2\beta^2(1-2\epsilon)^2}{\gamma_+ \gamma_-} \log \frac{\beta}{\eta}$. We hence get

$$\mathrm{LB2} \geq \begin{cases} -2n\,(1-2\epsilon)^2\,\alpha\beta \log \frac{\eta}{\alpha\beta} & \text{if } \alpha\beta < \eta \\ 0 & \text{if } \alpha\beta \geq \eta \end{cases}$$

Further relaxing the above condition by $\alpha\beta \log \frac{\eta}{\alpha\beta} \leq \eta - \alpha\beta \leq \eta$, we obtain:

$$\mathrm{LB2} \geq -2n\,(1-2\epsilon)^2\,\eta \tag{A.2.33}$$

Combining Equation (A.2.27), (A.2.30), and (A.2.33), we get

$$\Delta_{\mathrm{AUX}} \geq \frac{1}{16}\,(1-2\epsilon)^2\,\Delta_{\mathrm{EC}^2} - 2n\,(1-2\epsilon)^2\,\eta. \tag{A.2.34}$$

Hence, we have related $\Delta_{\mathrm{AUX}}(v \mid \psi)$ to $\Delta_{\mathrm{EC}^2,\psi}(v)$, as stated in Lemma 4.21.

Bounding Δ_{aux} against Δ_{ECED}

To finish the proof of Lemma 4.21, it remains to bound Δ_{AUX} against Δ_{ECED}. In this subsection, we complete the proof of Lemma 4.21, by showing that $\Delta_{\mathrm{AUX}}(v \mid \psi) + 2n\,(1-2\epsilon)^2\,\eta \geq \Delta_{\mathrm{ECED},\psi}(v)\,/64$.

Recall that ϵ is the noise rate of test v. Let $\rho = \frac{\epsilon}{1-\epsilon}$ be the discount factor for inconsistent root-causes. By the definition of Δ_{ECED} in Equation (4.5.2), we first expand the expected offset value of performing test v:

$$\mathbb{E}_{x_v}[\delta_{\text{OFFSET}}(x_v)] = \sum_{y_i \in \mathcal{Y}} (\alpha_i + \beta_i)(1 - \alpha_i - \beta_i)\epsilon \left(1 - \rho^2\right).$$

Denote $\gamma = \epsilon\left(1 - \rho^2\right)$. Then, we can expand Δ_{ECED} as

$$\begin{aligned}
\Delta_{\text{ECED}} = \sum_{y_i \in \mathcal{Y}} \Bigg(& \overbrace{(\alpha_i + \beta_i)(1 - \alpha_i - \beta_i)\left(1 - \gamma\right)}^{\text{(initial total edge weight)}-\text{(offset value)}} - \\
& \overbrace{(\gamma_+(\alpha_i + \rho\beta_i)(\alpha + \rho\beta - \alpha_i - \rho\beta_i) + \gamma_-(\beta_i + \rho\alpha_i)(\beta + \rho\alpha - \beta_i - \rho\alpha_i))}^{\text{expected remaining weight after discounting}} \Bigg) \\
= \gamma_+ \sum_{y_i \in \mathcal{Y}} & (-\gamma\alpha_i(\alpha - \alpha_i) + \alpha_i(\beta - \beta_i)(1 - \gamma - \rho) + \\
& \beta_i(\alpha - \alpha_i)(1 - \gamma - \rho) + \beta_i(\beta - \beta_i)(1 - \gamma - \rho^2)\Big) + \\
& \gamma_- \sum_{y_i \in \mathcal{Y}} (-\gamma\beta_i(\beta - \beta_i) + \beta_i(\alpha - \alpha_i)(1 - \gamma - \rho) + \\
& \alpha_i(\beta - \beta_i)(1 - \gamma - \rho) + \alpha_i(\alpha - \alpha_i)(1 - \gamma - \rho^2)\Big) \\
= \sum_{y_i \in \mathcal{Y}} & \Big(2(1 - \gamma - \rho)\alpha_i(\beta - \beta_i) + \left(\gamma_+(1 - \gamma - \rho^2) - \gamma_-\gamma\right)\beta_i(\beta - \beta_i) + \\
& \left(\gamma_-(1 - \gamma - \rho^2) - \gamma_+\gamma\right)\alpha_i(\alpha - \alpha_i)\Big)
\end{aligned}$$

Since $\gamma = \frac{\epsilon(1-2\epsilon)}{(1-\epsilon)^2}$, $1 - \gamma - \rho^2 = \frac{1-2\epsilon}{1-\epsilon}$, and $1 - \gamma - \rho = \left(\frac{1-2\epsilon}{1-\epsilon}\right)^2$, we have,

$$\begin{aligned}
\gamma_+(1 - \gamma - \rho^2) - \gamma_-\gamma &= (\alpha(1-\epsilon) + \beta\epsilon)\frac{1-2\epsilon}{1-\epsilon} - (\alpha\epsilon + \beta(1-\epsilon))\frac{\epsilon(1-2\epsilon)}{(1-\epsilon)^2} \\
&= \left(\frac{1-2\epsilon}{1-\epsilon}\right)^2 \alpha
\end{aligned}$$

Therefore

$$\begin{aligned}
\Delta_{\text{ECED}} &= \left(\frac{1-2\epsilon}{1-\epsilon}\right)^2 \left(\alpha \sum_{y_i \in \mathcal{Y}} \beta_i(\beta - \beta_i) + \beta \sum_{y_i \in \mathcal{Y}} \alpha_i(\alpha - \alpha_i) + 2 \sum_{y_i \in \mathcal{Y}} \alpha_i(\beta - \beta_i)\right) \\
&= \left(\frac{1-2\epsilon}{1-\epsilon}\right)^2 \Delta_{\text{EC}^2} \qquad \text{(A.2.35)}
\end{aligned}$$

Combining Equation (A.2.35) with Equation (A.2.34) we obtain

$$\begin{aligned}\Delta_{\text{AUX}} + 2n\,(1-2\epsilon)^2\,\eta &\geq \frac{(1-\epsilon)^2}{16}\Delta_{\text{ECED}} \\ &= \frac{1}{16}\,(1-2\epsilon)^2\,\Delta_{\text{EC}^2}\end{aligned}$$

With the results from Appendix §A.2.7 and §A.2.7, we therefore complete the proof of Lemma 4.21.

A.2.8 Proof of Theorem 4.19 Part 3: The Key Lemma Relating ECED to OPT

Bounding the Error Probability: Noise-free vs. Noisy Setting

Now that we have seen how ECED interacts with our auxiliary function regarding the one-step gain, it remains to understand how one can relate the one-step gain to the gain of an optimal policy $\Delta_{\text{AUX}}(\text{OPT} \mid \psi)$, over k steps. In this subsection, we make an important step towards this goal.

Specifically, we provide

Lemma A.7. *Consider a policy π of length k, and assume that we are using a stochastic estimator (SE). Let $p_{\text{E}}^{\top}$ be the error probability of SE before running policy π, $p_{\text{E},noisy}^{\perp}$ be the average error probability of SE after running π in the noisy setting, and $p_{\text{E},noiseless}^{\perp}$ be the average error probability of SE after running π in the noiseless setting. Then*

$$p_{\text{E},noiseless}^{\perp} \leq p_{\text{E},noisy}^{\perp}$$

Proof of Lemma A.7. Recall that a *stochastic estimator* predicts the value of a random variable, by randomly drawing from its distribution. Let π be a policy. We denote by $p_{\text{E}}(\pi_h)$ the expected error probability of an stochastic estimator after observing π_h :

$$p_{\text{E,noisy}}^{\perp} = \mathbb{E}_h[p_{\text{E}}(\pi_h)] = \sum_h p(\pi_h) \sum_{y \in \mathcal{Y}} p(y \mid \pi_h)(1 - p(y \mid \pi_h))$$

where $h \in \mathcal{V} \times \mathcal{X}$ denotes a set of test-outcome pairs, and π_h denotes a path taken by π, given that it observes h.

Now, let us see what happens in the noiseless setting: we run π exactly as it is, but in the end compute the error probability of the noiseless setting (i.e., as if we know

which test outcomes are corrupted by noise). Denote the noise put on the tests by Ξ, and the realized noise by ξ. We can imagine the noiseless setting through the following equivalent way: we ran the same policy π exactly as in the noisy setting. But upon completion of π we reveal what Ξ was. We thus have

$$p(y \mid \pi_h) = \sum_{\Xi=\xi} p(y \mid \pi_h, \xi) p(\xi \mid \pi)$$

The error probability upon observing π_h and $\Xi = \xi$ is

$$p_{\mathrm{E}}(\pi_h, \xi) = \sum_{y \in \mathcal{Y}} p(y \mid \pi_h, \xi)(1 - p(y \mid \pi_h, \xi)).$$

The expected error probability in the noiseless setting after running π is

$$p^{\perp}_{\mathrm{E,noiseless}} = \mathbb{E}_{h,n}[p_{\mathrm{E}}(\pi_h, \xi)] = \sum_{h,n} p(\pi_h, \xi) \sum_{y \in \mathcal{Y}} p(y \mid \pi_h, \xi)(1 - p(y \mid \pi_h, \xi)) \tag{A.2.36}$$

Now, we can relate $p^{\perp}_{\mathrm{E,noisy}}$ to $p^{\perp}_{\mathrm{E,noiseless}}$.

$$\begin{aligned}
p^{\perp}_{\mathrm{E,noisy}} &= \sum_{h} p(\pi_h) \sum_{y \in \mathcal{Y}} p(y \mid \pi_h)(1 - p(y \mid \pi_h)) \\
&= \sum_{h} p(\pi_h) \sum_{y \in \mathcal{Y}} \sum_{\xi} p(\xi \mid \pi_h) p(y \mid \pi_h, \xi)(1 - \sum_{n} p(\xi \mid \pi_h) p(y \mid \pi_h, \xi)) \\
&\overset{(a)}{\geq} \sum_{h} p(\pi_h) \sum_{y \in \mathcal{Y}} \sum_{\xi} p(\xi \mid \pi_h) p(y \mid \pi_h, \xi)(1 - p(y \mid \pi_h, \xi)) \\
&= \sum_{h,\xi} p(\pi_h, \xi) \sum_{y \in \mathcal{Y}} p(y \mid \pi_h, \xi)(1 - p(y \mid \pi_h, \xi))
\end{aligned}$$

where (a) is by Jensen's inequality and the fact that $f(x) = x(1-x)$ is concave. Combining with Equation (A.2.36) we complete the proof. □

Essentially, Lemma A.7 implies that, in terms of the reduction in the expected prediction error of SE, running a policy in the noise-free setting has higher gain than running the same policy in the noisy setting. This result is important to us, since analyzing a policy in the noise-free setting is often easier. We are going to use Lemma A.7 in the next section, to relate the gain of an optimal policy $\Delta_{\mathrm{EC}^2,\psi}(\mathrm{OPT})$ in the EC^2 objective (which assumes tests to be noise-free), with the gain $\Delta_{\mathrm{AUX}}(\mathrm{OPT} \mid \psi)$ in the auxiliary function (which considers noisy test outcomes).

The Key Lemma: One-step Gain of ECED VS. k-step Gain of OPT

Now we are ready to state our key lemma, which connects $\Delta_{\text{AUX}}(v \mid \psi)$ to $\Delta_{\text{AUX}}(\text{OPT} \mid \psi)$.

Lemma A.8 (Key Lemma). *Fix $\eta, \tau \in (0,1)$. Let $m = |\operatorname{supp}(\Theta)|$ be the number of root-causes, $n = |\mathcal{Y}|$ be the number of target values, $\text{OPT}(\delta_{\text{OPT}})$ be the optimal policy that achieves $p_{\text{ERR}}(\text{OPT}(\delta_{\text{OPT}})) \leq \delta_{\text{OPT}}$, and ψ_ℓ be the partial realization observed by running ECED with cost ℓ. We denote by $f^{avg}_{\text{AUX}}(\ell) := \mathbb{E}_{\psi_\ell}[f_{\text{AUX}}(\psi_\ell)]$ the expected value of $f_{\text{AUX}}(\psi_\ell)$ over all the paths ψ_ℓ at cost ℓ. Assume that $f^{avg}_{\text{AUX}}(\ell) \leq \delta_{\text{g}}$. We then have*

$$f^{avg}_{\text{AUX}}(\ell) - f^{avg}_{\text{AUX}}(\ell+1) \geq \frac{f^{avg}_{\text{AUX}}(\ell) - \delta_{\text{OPT}}}{k} \cdot \frac{c_\epsilon}{c_\delta} + c_{\eta,\epsilon}.$$

where $k \triangleq \text{cost}(\text{OPT}(\delta_{\text{OPT}}))$, $c_{\eta,\epsilon} \triangleq 2n(1-2\epsilon)^2\eta$, $c_\delta \triangleq (6c+8)\log(m/\delta_{\text{g}})$, $c \triangleq 8\left(\log(2m^2/\eta)\right)^2$, and $c_\epsilon \triangleq (1-2\epsilon)^2/16$.

Proof of Lemma A.8. Let ψ_ℓ be a path ending up at level ℓ of the greedy algorithm. Recall that $\Delta_{\text{EC}^2}(v \mid \psi_\ell)$ denotes the gain in f_{EC^2} if we perform test v and assuming it to be *noiseless* (i.e., we perform edge cutting as if the outcome of test v is noiseless), conditioned on partial observation ψ_ℓ. Further, recall that $\Delta_{\text{AUX}}(v \mid \psi_\ell)$ denotes the gain in f_{AUX} if we perform *noisy* test v after observing ψ_ℓ and perform Bayesian update on the root-causes.

Let $v = \arg\max_{v'} \Delta_{\text{ECED}}(v' \mid \psi_\ell)$ be the test chosen by ECED, and $\hat{v} = \arg\max_{v'} \Delta_{\text{EC}^2}(v' \mid \psi_\ell)$ be the test that maximizes Δ_{EC^2}, then by Lemma 4.21 we know

$$\begin{aligned}
\Delta_{\text{AUX}}(v \mid \psi_\ell) + c_{\eta,\epsilon} &\geq \frac{(1-\epsilon)^2}{16}\left(\Delta_{\text{ECED},\psi_\ell}(v)\right) \\
&\geq \frac{(1-\epsilon)^2}{16}\left(\Delta_{\text{ECED},\psi_\ell}(\hat{v})\right) \\
&= \frac{1}{16}(1-2\epsilon)^2\,\Delta_{\text{EC}^2,\psi_\ell}(\hat{v}) \qquad \text{(A.2.37)}
\end{aligned}$$

Note that $\Delta_{\text{EC}^2,\psi_\ell}(v)$ is the EC^2 gain of test v over the *normalized* edge weights at step $\ell+1$ in the noiseless setting. That is, upon observing ψ_ℓ, we create a new EC^2 problem instance (by considering the posterior probability over root-causes at ψ_ℓ), and run (noiseless) greedy algorithm w.r.t. the EC^2 objective on such problem instance. Recall that $c_\epsilon \triangleq (1-2\epsilon)/16$. By adaptive submodularity of f_{EC^2} (in the noiseless setting, see

Golovin, Krause, and Ray [GKR10b]), we obtain

$$\max_v \Delta_{\mathrm{EC}^2,\psi_\ell}(v) \overset{\substack{\text{adaptive}\\ \text{submodularity}}}{\geq} \frac{f^\top_{\mathrm{EC}^2,\psi_\ell} - \mathbb{E}\left[f^\perp_{\mathrm{EC}^2,\psi_\ell}\right]}{k}$$

where by $f^\top_{\mathrm{EC}^2,\psi_\ell}$ we mean the initial EC^2 objective value given partial realization ψ_ℓ, and by $\mathbb{E}\left[f^\perp_{\mathrm{EC}^2,\psi_\ell}\right]$ we mean the expected gain in f_{EC^2} when we run OPT (δ_{OPT}). Note that OPT (δ_{OPT}) has worst-case length k.

The above inequality has established a connection between $\Delta_{\mathrm{EC}^2,\psi}(v)$ and $\Delta_{\mathrm{EC}^2,\psi}(\mathrm{OPT})$. Now, imagine that we run the policy OPT (δ_{OPT}), and upon completion of the policy we can observe the noise. We consider the gain of such policy in f_{EC^2}:

$$f^\top_{\mathrm{EC}^2} - \mathbb{E}\left[f^\perp_{\mathrm{EC}^2}\right] \overset{\text{(a)}}{=} p^\top_{\mathrm{E}} - \mathbb{E}\left[f^\perp_{\mathrm{EC}^2}\right] \overset{\text{(b)}}{\geq} p^\top_{\mathrm{E}} - p^\perp_{\mathrm{E,noiseless}}.$$

The reason for step (a) is that the error probability of the stochastic estimator upon observing ψ_ℓ, i.e., $p^\top_{\mathrm{E}}$, is equivalent to the total amount of edge weight at ψ_ℓ, i.e., $f^\top_{\mathrm{EC}^2,\psi_\ell}$. The reason for step (b) is that under the noiseless setting (i.e., assuming we have access to the noise), the EC^2 objective is always a lower-bound on the error probability of the *stochastic estimator* (due to normalization). Thus, $\mathbb{E}\left[f^\perp_{\mathrm{EC}^2}\right] \leq p^\perp_{\mathrm{E,noiseless}}$.

Hence we get

$$\Delta_{\mathrm{AUX}}(v \mid \psi_\ell) + c_{\eta,\epsilon} \geq c_\epsilon \frac{p^\top_{\mathrm{E},\psi_\ell} - p^\perp_{\mathrm{E,noiseless},\psi_\ell}}{k}.$$

Here $p^\top_{\mathrm{E},\psi_\ell}$ denotes the error probability under $\mathbb{P}[Y \mid \psi_\ell]$, and $p^\perp_{\mathrm{E,noisy},\psi_\ell}$ denotes the expected error probability of running OPT (δ_{OPT}) after ψ_ℓ in the *noise-free* setting. By Lemma A.7 we get

$$\Delta_{\mathrm{AUX}}(v \mid \psi_\ell) + c_{\eta,\epsilon} \geq c_\epsilon \frac{p^\top_{\mathrm{E},\psi_\ell} - p^\perp_{\mathrm{E,noisy},\psi_\ell}}{k},$$

where $p^\perp_{\mathrm{E,noisy},\psi_\ell}$ denotes the expected error probability of running OPT (δ_{OPT}) after ψ_ℓ in the *noisy* setting. By (the lower bound in) Lemma A.4, we know that $p^\top_{\mathrm{E},\psi_\ell} = p_{\mathrm{E}}(\psi_\ell) \geq p^{\mathrm{MAP}}_{\mathrm{ERR}}(\psi_\ell)$, and hence

$$\Delta_{\mathrm{AUX}}(v \mid \psi_\ell) + c_{\eta,\epsilon} \geq c_\epsilon \frac{p^{\mathrm{MAP}}_{\mathrm{ERR}}(\psi_\ell) - \delta_{\mathrm{OPT}}}{k},$$

Taking expectation on ψ_ℓ, we get

$$\mathbb{E}_{\psi_\ell}\left[\Delta_{\text{AUX}}(v \mid \psi_\ell) + c_{\eta,\epsilon}\right] \geq c_\epsilon \frac{\mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right] - \delta_{\text{OPT}}}{k}. \tag{A.2.38}$$

Using (the upper bound in) Lemma 4.20, we obtain

$$\begin{aligned} f_{\text{AUX}}^{\text{avg}}(\ell) &= \mathbb{E}_{\psi_\ell}[f_{\text{AUX}}(\psi_\ell)] \\ &\leq (3c+4)\left(\mathbb{E}_{\psi_\ell}\left[\mathbb{H}_2\left(p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right)\right] + \mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right]\log m\right) \\ &\overset{\text{(a)}}{\leq} (3c+4)\left(\mathbb{H}_2\left(\mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right]\right) + \mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right]\log m\right) \end{aligned} \tag{A.2.39}$$

where (a) is by Jensen's inequality.

Suppose we run ECED, and achieve expected error probability δ_{g}, then clearly before ECED terminates we have $\mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right] \geq \delta_{\text{g}}$. Assuming $\mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right] \leq 1/2$, we have

$$\begin{aligned} f_{\text{AUX}}^{\text{avg}}(\ell) &\leq (3c+4)\mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right]\left(2\log\frac{1}{\mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right]} + \log m\right) \\ &\leq (3c+4)\mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right]\left(2\log\frac{1}{\delta_{\text{g}}} + \log m\right) \\ &\leq \mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right]\cdot(6c+8)\log\frac{m}{\delta_{\text{g}}} \end{aligned} \tag{A.2.40}$$

which gives us

$$\mathbb{E}_{\psi_\ell}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_\ell)\right] \geq \frac{f_{\text{AUX}}^{\text{avg}}(\ell)}{(6c+8)\log\frac{m}{\delta_{\text{g}}}} \overset{c_\delta \triangleq (6c+8)\log\frac{m}{\delta_{\text{g}}}}{=} \frac{f_{\text{AUX}}^{\text{avg}}(\ell)}{c_\delta}. \tag{A.2.41}$$

Combining Equation (A.2.41) with Equation (A.2.38), we get

$$\begin{aligned} f_{\text{AUX}}^{\text{avg}}(\ell) - f_{\text{AUX}}^{\text{avg}}(\ell+1) &= \mathbb{E}_{\psi_\ell}[\Delta_{\text{AUX}}(v \mid \psi_\ell)] \\ &\geq c_\epsilon \frac{\frac{f_{\text{AUX}}^{\text{avg}}(\ell)}{c_\delta} - \delta_{\text{OPT}}}{k} - c_{\eta,\epsilon} \\ &= \frac{f_{\text{AUX}}^{\text{avg}}(\ell) - \delta_{\text{OPT}} c_\delta}{k}\cdot\frac{c_\epsilon}{c_\delta} - c_{\eta,\epsilon} \end{aligned}$$

which completes the proof. □

A.2.9 Proof of Theorem 4.19 Final Step: Near-optimality of ECED

We are going to put together the pieces from the previous subsections, to give a proof of our main theoretical result (Theorem 4.19).

Proof of Theorem 4.19. In the following, we use both $\mathrm{OPT}_{[k]}$ and $\mathrm{OPT}(\delta_{\mathrm{OPT}})$ to represent the optimal policy that achieves prediction error δ_{OPT}, with worst-case cost (i.e., length) k. Define $S(\pi, h)$ to be the (partial) realization seen by policy π under realization/ hypothesis h. With slight abuse of notation, we use $f_{\mathrm{AUX}}^{\mathrm{avg}}\left(\mathrm{OPT}_{[k]}\right) := \mathbb{E}_h\left[f_{\mathrm{AUX}}(S(\mathrm{OPT}_{[k]}, h))\right]$ to denote the expected value achieved by running $\mathrm{OPT}_{[k]}$.

After running $\mathrm{OPT}_{[k]}$, we know by Lemma 4.20 that the expected value of f_{AUX} is lower bounded by $2c \cdot \delta_{\mathrm{OPT}}$. That is, $\delta_{\mathrm{OPT}} \cdot c_\delta \le f_{\mathrm{AUX}}^{\mathrm{avg}}\left(\mathrm{OPT}_{[k]}\right) \cdot \frac{c_\delta}{2c} \le f_{\mathrm{AUX}}^{\mathrm{avg}}\left(\mathrm{OPT}_{[k]}\right) \cdot 4\log(n/\delta_{\mathrm{g}})$, where the last inequality is due to $c_\delta \triangleq (6c+8)\log\frac{m}{\delta_{\mathrm{g}}} < 8c\log\frac{m}{\delta_{\mathrm{g}}}$. We then have

$$\begin{aligned} f_{\mathrm{AUX}}^{\mathrm{avg}}(\ell) - f_{\mathrm{AUX}}^{\mathrm{avg}}(\ell+1) &\overset{\text{Lemma A.8}}{\ge} \left(f_{\mathrm{AUX}}^{\mathrm{avg}}(\ell) - \delta_{\mathrm{OPT}} \cdot c_\delta\right) \cdot \frac{c_\epsilon}{k c_\delta} - c_{\eta,\epsilon} \\ &\ge \left(f_{\mathrm{AUX}}^{\mathrm{avg}}(\ell) - f_{\mathrm{AUX}}^{\mathrm{avg}}\left(\mathrm{OPT}_{[k]}\right) \cdot 4\log\frac{m}{\delta_{\mathrm{g}}}\right) \cdot \frac{c_\epsilon}{k c_\delta} - c_{\eta,\epsilon} \end{aligned} \tag{A.2.42}$$

Let $\Delta_\ell \triangleq f_{\mathrm{AUX}}^{\mathrm{avg}}(\ell) - f_{\mathrm{AUX}}^{\mathrm{avg}}\left(\mathrm{OPT}_{[k]}\right) \cdot 4\log\frac{m}{\delta_{\mathrm{g}}}$, so that Inequality (A.2.42) implies $\Delta_\ell - \Delta_{\ell+1} \ge \Delta_\ell \cdot \frac{c_\epsilon}{k c_\delta} - c_{\eta,\epsilon}$. From here we get $\Delta_{\ell+1} \le \left(1 - \frac{c_\epsilon}{k c_\delta}\right)\Delta_\ell + c_{\eta,\epsilon}$, and hence

$$\begin{aligned} \Delta_{k'} &\le \left(1 - \frac{c_\epsilon}{k c_\delta}\right)^{k'} \Delta_0 + \sum_{i=0}^{k'} \left(1 - \frac{c_\epsilon}{k c_\delta}\right)^{i} \cdot c_{\eta,\epsilon} \\ &\overset{(a)}{\le} \exp\left(-k' \frac{c_\epsilon}{k c_\delta}\right) \Delta_0 + \frac{1 - \left(1 - \frac{c_\epsilon}{k c_\delta}\right)^{k'}}{\frac{c_\epsilon}{k c_\delta}} \cdot c_{\eta,\epsilon} \\ &\overset{(b)}{\le} \exp\left(-k' \frac{c_\epsilon}{k c_\delta}\right) \Delta_0 + \frac{k c_\delta}{c_\epsilon} \cdot c_{\eta,\epsilon} \end{aligned}$$

where step (a) is because $(1-x)^{k'} \le \exp(-k'x)$ for any $x < 1$, and step (b) is due to

$\left(1-\frac{c_\epsilon}{kc_\delta}\right)^{k'} > 0$. It follows that

$$\begin{aligned} & f_{\text{AUX}}^{\text{avg}}(k') - f_{\text{AUX}}^{\text{avg}}\left(\text{OPT}_{[k]}\right)\cdot 4\log\frac{m}{\delta_{\text{g}}} \\ & \quad \le \exp\left(-k'\frac{c_\epsilon}{kc_\delta}\right)\Delta_0 + \frac{kc_\delta}{c_\epsilon}\cdot c_{\eta,\epsilon} \\ & \quad \le \exp\left(-k'\frac{c_\epsilon}{kc_\delta}\right)\left(f_{\text{AUX}}^{\text{avg}}(\emptyset) - f_{\text{AUX}}^{\text{avg}}\left(\text{OPT}_{[k]}\right)\cdot 4\log\frac{m}{\delta_{\text{g}}}\right) + \frac{kc_\delta}{c_\epsilon}\cdot c_{\eta,\epsilon} \end{aligned}$$

This gives us

$$\begin{aligned} & f_{\text{AUX}}^{\text{avg}}(k') \\ & \le \underbrace{f_{\text{AUX}}^{\text{avg}}(\emptyset)\cdot\exp\left(-k'\frac{c_\epsilon}{kc_\delta}\right)}_{\text{UB1}} + \underbrace{f_{\text{AUX}}^{\text{avg}}\left(\text{OPT}_{[k]}\right)\cdot 4\log\frac{m}{\delta_{\text{g}}}\left(1-\exp\left(-k'\frac{c_\epsilon}{kc_\delta}\right)\right)}_{\text{UB2}} + \underbrace{\frac{kc_\delta}{c_\epsilon}\cdot c_{\eta,\epsilon}}_{\text{UB3}} \end{aligned} \tag{A.2.43}$$

Denote the three terms on the RHS. of Equation (A.2.43) as UB1, UB2 and UB3, respectively. We get

$$\begin{cases} \text{UB1} & \overset{\text{Eq (A.2.39)}}{\le} (3c+4)(1+\log m)\cdot\exp\left(-k'\frac{c_\epsilon}{kc_\delta}\right) \\ \text{UB2} & \overset{\text{Eq (A.2.40)}}{<} (6c+8)\cdot\delta_{\text{OPT}}\log\frac{m}{\delta_{\text{OPT}}}\cdot 4\log\frac{m}{\delta_{\text{g}}} \\ \text{UB3} & = k\cdot(6c+8)\log\frac{m}{\delta_{\text{g}}}\cdot\frac{2n(1-2\epsilon)^2\eta}{\frac{1}{16}(1-2\epsilon)^2} = (6c+8)\cdot 32\cdot k\cdot\log\frac{m}{\delta_{\text{g}}}\cdot n\eta \end{cases}$$

Now we set

$$\begin{cases} k' & \triangleq \frac{kc_\delta}{c_\varepsilon}\cdot\ln\frac{8\log m}{\delta_{\text{g}}} \\ \delta_{\text{OPT}} & \triangleq \frac{\delta_{\text{g}}}{64\cdot 36\cdot\log m\cdot\log\frac{1}{\delta_{\text{g}}}\cdot\log\frac{m}{\delta_{\text{g}}}} \end{cases} \tag{A.2.44}$$

and obtain $\exp\left(-k'\frac{c_\epsilon}{kc_\delta}\right) = \frac{\delta_{\text{g}}}{8\log m}$. It is easy to verify that UB1 $\le 2c\cdot\frac{\delta_g}{4}$, and UB2 $\le 2c\cdot\frac{\delta_g}{2}$.

We further set

$$\eta \triangleq \frac{\delta_{\text{g}}}{16\cdot 32\cdot kn\cdot\log\frac{m}{\delta_{\text{g}}}}, \tag{A.2.45}$$

and obtain UB3 $= 2c\cdot\frac{\delta_g}{4}$.

Combining the upper bound derived above for UB1, UB2, UB3, and by Equation (A.2.43), we get $f_{\text{AUX}}^{\text{avg}}(k') \leq 2c \cdot \delta_{\text{g}}$. By Lemma 4.20 we know that the error probability is upper bounded by $p_{\text{ERR}} = \mathbb{E}_{\psi_{k'}}\left[p_{\text{ERR}}^{\text{MAP}}(\psi_{k'})\right] \leq \frac{f_{\text{AUX}}^{\text{avg}}(k')}{2c} \leq \delta_{\text{g}}$. That is, with the cost k' specified in Equation (A.2.44), ECED is guaranteed to achieve $p_{\text{ERR}} \leq \delta_{\text{g}}$.

It remains to compute the (exact) value of k'. Combining the definition of $c \triangleq 8\left(\log(2m^2/\eta)\right)^2$ and $c_\delta \triangleq (6c+8)\log(m/\delta_{\text{g}})$ with Equation (A.2.45) it is easy to verify that

$$c_\delta \leq c_1 \cdot \left(\log \frac{mk}{\delta_{\text{g}}}\right)^2 \cdot \log \frac{m}{\delta_{\text{g}}},$$

holds for some constant c_1. Therefore by Equation (A.2.44),

$$k' \leq k \cdot c_1 \left(\log \frac{mk}{\delta_{\text{g}}}\right)^2 \log \frac{m}{\delta_{\text{g}}} \cdot \frac{1}{c_\varepsilon} \ln \frac{8 \log m}{\delta_{\text{g}}} = O\left(\frac{k}{c_\varepsilon}\left(\log \frac{mk}{\delta_{\text{g}}}\right)^2 \left(\log \frac{m}{\delta_{\text{g}}}\right)^2\right).$$

To put it in words, it suffices to run ECED for $O\left(\frac{k}{c_\varepsilon}\left(\log \frac{mk}{\delta_{\text{g}}}\right)^2 \left(\log \frac{m}{\delta_{\text{g}}}\right)^2\right)$ steps to have expected error below δ_{g}, where k denotes the worst-case cost the optimal policy that achieves expected error probability $\delta_{\text{OPT}} \triangleq O\left(\frac{\delta_{\text{g}}}{(\log m \cdot \log(1/\delta_{\text{g}}))^2}\right)$; hence the completion of the proof. □

A.3 Proofs from Chapter 5

A.3.1 Proof of Lemma 5.2: Adaptive Submodularity of f_{Det}

Proof of Lemma 5.2 . We first decompose each voting element into a set of voting elements, each carrying equal weights for all of its outgoing edges. Let the new voting elements set be Σ'. Then for all $\sigma \in \Sigma'$ and $v_1, v_2 \in \mathcal{V}$, it holds that

$$(w_{\sigma,v_1} > 0) \wedge (w_{\sigma,v_2} > 0) \Rightarrow w_{\sigma,v_1} = w_{\sigma,v_2}.$$

We then show that for all $\sigma \in \Sigma'$ and $v \in \mathcal{V}$, the function $f_{\sigma,v}$ is monotone submodular for edges in the new bipartite graph.

Let $\Delta_f((v^*, x_{v^*}) \mid \mathbf{x}_\mathcal{A}) \triangleq f_{\sigma,v}(\mathbf{x}_\mathcal{A} \cup \{(v^*, x_{v^*})\}) - f_{\sigma,v}(\mathbf{x}_\mathcal{A})$ be the marginal gain of $f_{\sigma,v}$ over set $\mathbf{x}_\mathcal{A}$ by selecting (v^*, x_{v^*}). We need to show that for each $\mathbf{x}_\mathcal{A} \subseteq \mathbf{x}_\mathcal{B} \subseteq \mathcal{V} \times \mathcal{X}$, and $(v^*, x_{v^*}) \in \mathcal{V} \times \mathcal{X}$ it holds that

$$\Delta_f((v^*, x_{v^*}) \mid \mathbf{x}_\mathcal{A}) \geq \Delta_f((v^*, x_{v^*}) \mid \mathbf{x}_\mathcal{B}) \geq 0.$$

The proof falls naturally into three parts.

- If $x_{v^*} = +1$, then $g(\sigma, v, \mathbf{x}_{\mathcal{A}\cup\{v^*\}}) = g(\sigma, v, \mathbf{x}_\mathcal{A}) \leq 1$. Let $\beta_\mathcal{A} = 1 - g(\sigma, v, \mathbf{x}_\mathcal{A})$. We have

$$\begin{aligned}
\Delta_f((v^*, +1) \mid \mathbf{x}_\mathcal{A}) &= f_{\sigma,v}(\mathbf{x}_\mathcal{A} \cup \{(v^*, +1)\}) - f_{\sigma,v}(\mathbf{x}_\mathcal{A}) \\
&= \min\left\{\max\left(\max_{v':(v',+1)\in\mathbf{x}_\mathcal{A}} w_{\sigma,v'},\ w_{\sigma,v^*}\right),\ \beta_\mathcal{A} \cdot w_{\sigma,v}\right\} \\
&\quad - \min\left\{\max_{v':(v',+1)\in\mathbf{x}_\mathcal{A}} w_{\sigma,v'},\ \beta_\mathcal{A} \cdot w_{\sigma,v}\right\} \\
&= \begin{cases} 0 & \text{if } \exists v' : (v', +1) \in \mathbf{x}_\mathcal{A} \wedge w_{\sigma,v'} > 0; \\ \min\left(w_{\sigma,v^*},\ \beta_\mathcal{A} \cdot w_{\sigma,v}\right) & \text{otherwise.} \end{cases}
\end{aligned}$$

  For both cases on the RHS, it holds that $\Delta((v^*, +1) \mid \mathbf{x}_\mathcal{A}) \geq \Delta((v^*, +1) \mid \mathbf{x}_\mathcal{B}) \geq 0$.

- If $x_{v^*} = -1$, and $\exists v' : (v', +1) \in \mathbf{x}_\mathcal{A} \wedge w_{\sigma,v'} > 0$, then the edge (σ, v) is fully covered by $\mathbf{x}_\mathcal{A}$. Thus $\Delta_f((v^*, -1) \mid \mathbf{x}_\mathcal{A}) = \Delta_f((v^*, -1) \mid \mathbf{x}_\mathcal{B}) = 0$.

- Otherwise, $\max_{v':(v',+1)\in\mathbf{x}_{\mathcal{A}}} w_{\sigma,v'} = 0$. Therefore,

$$\begin{aligned}
&\Delta_f((v^*,-1) \mid \mathbf{x}_{\mathcal{A}}) \\
&= w_{\sigma,v} \cdot \Big(g(\sigma, v, \mathbf{x}_{\mathcal{A}\cup\{v^*\}}) - g(\sigma, v, \mathbf{x}_{\mathcal{A}})\Big) \\
&\geq w_{\sigma,v} \cdot \Big(g(\sigma, v, \mathbf{x}_{\mathcal{B}\cup\{v^*\}}) - g(\sigma, v, \mathbf{x}_{\mathcal{B}})\Big) \\
=&\Delta_f((v^*,-1) \mid \mathbf{x}_{\mathcal{B}})
\end{aligned}$$

  The inequality holds because g is constructed as a concave function. Since g is also non-decreasing, we have $\Delta_f((v^*,-1) \mid \mathbf{x}_{\mathcal{A}}) \geq 0$.

By definition $f_{\textsc{Det}}$ is a non-negative linear combination of monotone submodular functions, and hence is also monotone submodular. □

A.4 Proofs from Chapter 6

A.4.1 Proof of Theorem 6.6: Adaptivity Gap under Cardinality Constraint $k = 2$

To prove Theorem 6.6, we first establish the following Lemma:

Lemma A.9. *Let $\mathcal{V} = \{1, \dots, t\}$. Let $(f, \mathbb{P})$ be adaptive submodular. Then, let $f'(\mathbf{x}_{\mathcal{A}}) \triangleq [f(\mathbf{x}_{\mathcal{A}}) \mid x_1]$, that is, f' is the conditional expectation of f conditioned on the observation of test 1. Then f' is also adaptive submodular. Further, the expected utilities of f and f' are the same.*

Proof. The conditional expectation is a linear operator, and linear combinations of adaptive submodular functions are adaptive submodular. By the definition of f' and the use of the tower rule, we get that f ad f' have the same expected value. □

Let us assume WLOG that the optimal policy for $(f, \mathbb{P})$ starts with choosing element 1.

Lemma A.10. $\textsc{gap}_k(f, \mathbb{P}) = \textsc{gap}_k(f', \mathbb{P})$.

Proof. First, $\textsc{opt}_{const}$ are equal for the two functions, following from Lemma A.9.

Second, any policy that starts with performing test 1 will have the same expected utility on f and f' (given $k = 2$). For policies that do not start with 1, the expected utility for f' must be at most that of f, since the policy gets less information for f'. Thus, since we assumed that for f the optimal policy starts with 1, the optimal policy for f' must remain the same as for f, getting the same expected utility. □

It is obvious that, for f', expected utilities do not depend on observations of elements other than 1. Thus, we can "remove all other observations". Let the distribution $\mathbb{P}'$ defined such that the marginal for 1 is the same as $\mathbb{P}$ and for all other elements $a \neq 1$, $\mathbb{P}[(a, *)] = 1$ for some arbitrary fixed observation $* \in O$. Then, based on the above, $\textsc{gap}_2(f', \mathbb{P}) = \textsc{gap}_2(f', \mathbb{P}')$. Now we are ready to prove the upper bound on the adaptivity gap for cardinality constraint 2.

Proof of Theorem 6.6. From the above lemmas, we know that it is enough to deal with distributions for which only one observation is not deterministic. But such a distribution

always factorizes (that is, the distributions for each observation are independent), and thus we can use the theorem of [AN15], that states that the adaptivity gap is at most $\frac{e}{e-1}$ in the case of factorizing distributions. □

A.4.2 Proof of Theorem 6.7: Adaptivity Gap under Cardinality Constraint k

Gap for Cardinality Constraint that is a Power of 2

In this subsection, we show an upper bound on the adaptivity gap for any cardinality constraint $k = 2^l$. We will use induction on l and start with the result of the previous section, that is, we assume for $k = 2$, an upper bound on the gap is $\textsc{gap}_2$.

Now we consider the case when we have a set $\mathcal{V}$ and an adaptive submodular function f over $\mathcal{V}, \mathcal{X}$, and distribution $\mathbb{P}$. We construct the set $\mathcal{V}_{l-1}$: every element in this set is a policy of depth 2^{l-1} on the original problem $(\mathcal{V}, \mathcal{X}, \mathbb{P})$. We define the observation set $\mathcal{X}_{l-1}$ as vectors of observations of length 2^{l-1} and the distribution $\mathbb{P}_{l-1}$ induced by $\mathbb{P}$. We also define the (adaptive submodular) function over $(\mathcal{V}_{l-1}, \mathcal{X}_{l-1}, \mathbb{P}_{l-1})$ the expected utility induced by the original problem.

It is easy to see that a policy of depth 2 over the new problem is essentially a policy of depth 2^l over the original problem.

Let the subset $\mathcal{V}'_{l-1}$ of $\mathcal{V}_{l-1}$ consist of "constant" policies of depth 2^{l-1}, that is, all the subsets of $\mathcal{V}$ of cardinality 2^{l-1}. A set of two elements from this subset is equivalent to a set of at most 2^l elements from $\mathcal{V}$.

We introduce the following notations:

$\textsc{opt}_{\text{s}}^{\text{tree}}$	the expected utility of the optimal sequential policy that chooses two elements from $\mathcal{V}_{l-1}$ sequentially
$\textsc{opt}_{\text{c}}^{\text{tree}}$	the expected utility of the optimal constant policy that chooses two elements from $\mathcal{V}_{l-1}$ without observing anything.
$\textsc{opt}_{\text{c}}^{\text{set}}$	the expected utility of the optimal constant policy that chooses two elements from $\mathcal{V}'_{l-1}$, without observing anything.

What we want to upper bound is the ratio

$$\frac{\mathrm{OPT}_{\mathrm{s}}^{\mathrm{tree}}}{\mathrm{OPT}_{\mathrm{c}}^{\mathrm{set}}} = \mathrm{GAP}_k \,.$$

By the definition of GAP_2 we know that

$$\frac{\mathrm{OPT}_{\mathrm{s}}^{\mathrm{tree}}}{\mathrm{OPT}_{\mathrm{c}}^{\mathrm{tree}}} \leq \mathrm{GAP}_2 \,. \tag{A.4.1}$$

In the rest of this subsection, we are going to upper bound the term

$$\frac{\mathrm{OPT}_{\mathrm{c}}^{\mathrm{tree}}}{\mathrm{OPT}_{\mathrm{c}}^{\mathrm{set}}} \,. \tag{A.4.2}$$

We will use the result of Goundan and Schulz [GS07] on approximate greedy submodular optimization. Theorem 1 of Goundan and Schulz [GS07] applied to $k = 2$ states that the ratio of an approximate greedy policy that selects elements whose expected marginal gain is at least α time the expected marginal gain of the optimal choice, we get that the ratio between the expected utility of the optimal policy and that of the approximal greedy policy is at most $\frac{e^{1/\alpha}}{e^{1/\alpha}-1} \geq 1 + \alpha$.

Let us approximate $\mathrm{OPT}_{\mathrm{c}}^{\mathrm{set}}$ by greedily choosing two elements from $\mathcal{V}'_{l-1}$. We can view this greedy algorithm as an approximate greedy algorithm for finding $\mathrm{OPT}_{\mathrm{c}}^{\mathrm{tree}}$ that is not allowed to pick elements from $\mathcal{V}_{l-1} \setminus \mathcal{V}'_{l-1}$:

1. To approximate the first element the optimal sequential algorithm picks, we choose the one from $\mathcal{V}'_{l-1}$ with the highes expected utility. From the induction hypothesis, we know that this choice is a $\mathrm{GAP}_{k/2}$-approximation of the optimal choice.

2. To choose the second element, we exclude all the elements from $\mathcal{V}_{l-1}$ whose top choice is equal to the top choice of the first element from the optimal sequential policy. The residual base set is still a set of decision trees of depth 2^{l-1}, but those trees do not contain the excluded "primitive" element. Choosing the best constant tree form this is a $\mathrm{GAP}_{k/2}$-approximation of choosing the decision tree with the highest marginal gain.

The above approximate greedy algorithm shows that one can have a set of two constant decision trees such that their expected utility is a $(1 + \mathrm{GAP}_{k/2})$-approximation of the

optimal choice of two arbitrary trees. Thus,

$$\frac{\text{OPT}_c^{\text{tree}}}{\text{OPT}_c^{\text{set}}} \leq 1 + \text{GAP}_{k/2}\,.$$

Reading (A.4.1) and (A.4.2) together we get

$$\begin{aligned}
\text{GAP}_k &\leq \text{GAP}_2(1 + \text{GAP}_{k/2}) \\
&\leq \text{GAP}_2(1 + (\text{GAP}_2(1 + \text{GAP}_{k/4})) \\
&\cdots \\
&\leq \text{GAP}_2(1 + (\text{GAP}_2(1 + \cdots \text{GAP}_2(1 + \text{GAP}_2) \cdots)) \\
&= \sum_{i=1}^{\log k} \text{GAP}_2^i = \frac{\text{GAP}_2^{\log k+1} - 1}{\text{GAP}_2 - 1} \\
&\leq \frac{\text{GAP}_2}{\text{GAP}_2 - 1} \text{GAP}_2^{\log k} \\
&= \frac{\text{GAP}_2}{\text{GAP}_2 - 1} k^{\log(\text{GAP}_2)}
\end{aligned} \tag{A.4.3}$$

Gap for Arbitrary Cardinality Constraint

From the previous part, we know that $\text{GAP}_k \leq \frac{\text{GAP}_2}{\text{GAP}_2 - 1} k^{\log(\text{GAP}_2)}$ for cardinality constraints that are a power of 2. Now we turn our attention to general cardinality constraints. The bound will be an immediate implication of the previous sections and the following lemma:

Lemma A.11. *Let $k < k' < 2k$. Then, $\text{OPT}_{seq}(k') \leq 2\text{OPT}_{seq}(k)$.*

Proof of Theorem 6.7. Let us denote the optimal policy's expected utility after the first k steps by $\text{OPT}_{\text{seq}}(k')\mid_k$. Then,

$$\text{OPT}_{\text{seq}}(k') = \text{OPT}_{\text{seq}}(k')\mid_k + (\text{OPT}_{\text{seq}}(k) - \text{OPT}_{\text{seq}}(k')\mid_k)\,.$$

Our first observation is that $\text{OPT}_{\text{seq}}(k')\mid_k$ can not be greater than $\text{OPT}_{\text{seq}}(k)$. Indeed, if that were the case, then the first k steps of the optimal policy for k' would work better in the first k steps than the optimal policy for k itself!

The second observation is that $\text{OPT}_{\text{seq}}(k) - \text{OPT}_{\text{seq}}(k')\mid_k$ can not be greater than $\text{OPT}_{\text{seq}}(k)$ either. To see this, we have to use the adaptive submodular property. The term $\text{OPT}_{\text{seq}}(k) - \text{OPT}_{\text{seq}}(k')\mid_k$ is the conditional expectation of the marginal gain of the last

$k' - k$ elements picked by the optimal policy. Implied by the diminishing returns condition, this expected gain can not be more than if we picked the last $k' - k$ elements first. The expected utility of these elements can not be more than $\text{OPT}_{\text{seq}}(k)$, for the same reason as in the previous paragraph.

Putting these together, we get

$$\begin{aligned}\text{OPT}_{\text{seq}}(k') &= \text{OPT}_{\text{seq}}(k')\mid_k + (\text{OPT}_{\text{seq}}(k) - \text{OPT}_{\text{seq}}(k')\mid_k) \\ &\leq \text{OPT}_{\text{seq}}(k) + \text{OPT}_{\text{seq}}(k) \leq 2\text{OPT}_{\text{seq}}(k)\,.\end{aligned}$$

Hence it completes the proof. □

Proof of Theorem A.4.2. Eq. (A.4.3) provides an upper bound on the adaptivity gap for cardinality constraints that are a power of 2, i.e., $\text{GAP}_k \leq \frac{\text{GAP}_2}{\text{GAP}_2 - 1} k^{\log(\text{GAP}_2)}$ It follows from Lemma A.11 that for any cardinality constraint k, the adaptivity gap under general cardinality constraints k can be upper bounded as $\text{GAP}_k \leq \frac{2\text{GAP}_2}{\text{GAP}_2 - 1} k^{\log(\text{GAP}_2)}$. □

A.4.3 Proof of Theorem 6.8: Bounding BatchGreedy against $\pi^*_{\text{batch},k}$

Theorem 6.8 rests on the following Lemma, which allows to view the selection of batches as items in a modified problem instance.

Lemma A.12. *Let $\mathcal{V} = \{1, \ldots, t\}$, $\mathcal{X}$ be finite sets; $f : 2^{\mathcal{V}\times\mathcal{X}} \to \mathbb{N}$ monotonic and submodular, and $P(\mathbf{X}_\mathcal{V})$ such that $(f, \mathbb{P})$ is adaptive submodular. Let ℓ be some fixed integer. Let $\mathcal{B}_1, \ldots, \mathcal{B}_s \subseteq \mathcal{V}$ be subsets of size ℓ of the groundset $\mathcal{V}$, and define, for $i \in \{1, \ldots, s\}$, $\mathbf{Z}_i = \mathbf{X}_{\mathcal{B}_i}$ as the random outcome of the set $\mathcal{B}_i$. Let $\mathcal{W} = \{1, \ldots, s\}$ and $\mathbb{P}'(\mathbf{Z}_\mathcal{W})$ be the distribution over $\mathbf{Z}_1, \ldots, \mathbf{Z}_s$ induced by $\mathbb{P}$. Let $\mathcal{X}' = \bigcup_{i\in\mathcal{W}} \text{range}(\mathbf{Z}_i)$. Define the function*

$$\phi : 2^{\mathcal{W}\times\mathcal{X}'} \to 2^{\mathcal{V}\times\mathcal{X}}, \phi(\{(a_1, \mathbf{z}_1), \ldots, (a_r, \mathbf{z}_r)\}) = \bigcup_{j=1}^{r} \{(v, x) : v \in \mathcal{B}_j, x = [\mathbf{z}_j]_v\}$$

and define $g : 2^{\mathcal{W}\times\mathcal{X}'} \to \mathbb{N}$ by $g(\mathcal{S}) = f(\phi(\mathcal{S}))$. Then g is submodular, and $(g, \mathbb{P}')$ is adaptive submodular.

Proof. Submodularity of g is immediate (see, e.g., Nemhauser, Wolsey, and Fisher [NWF78]). In order to prove adaptive submodularity, fix element $i \in \mathcal{W}$, corresponding

to $\mathcal{B}_i$, and let $\mathcal{A} \subseteq \mathcal{W} \times \mathcal{X}'$. Since $\mathbb{P}'(\mathcal{A}) > 0$, $\phi(\mathcal{A})$ cannot contain two elements (v, x_1) and (v, x_2) with $x_1 \neq x_2$.

Consider the marginal gain,

$$\Delta_g(i \mid \mathcal{A}) = \sum_{\mathbf{z}} \mathbb{P}'(\mathbf{z}_i \mid \mathcal{A}) \Big[f(\phi(\mathcal{A} \cup (i, \mathbf{z}_i)) - f(\phi(\mathcal{A})) \Big],$$

and let

$$\hat{\phi} \triangleq \phi(\mathcal{A} \cup (i, \mathbf{z}_i)) \setminus \phi(\mathcal{A}) = \{(i_1, x_1), \ldots, (i_\ell, x_\ell)\}.$$

Further let $\hat{\phi}_j \triangleq \{(i_1, x_1), \ldots, (i_j, x_j)\}$. It then holds that

$$\Delta_g(i \mid \mathcal{A}) = \sum_{j=1}^{\ell} \mathbb{E}_{\hat{\phi}_{j-1}} \left[\Delta_f(i_j \mid \phi(\mathcal{A}) \cup \hat{\phi}_{j-1}) \mid \phi(\mathcal{A}) \right].$$

Adaptive submodularity of $(g, \mathbb{P}')$ now follows from the adaptive submodularity of f, since $\Delta_f(i_j \mid \phi(\mathcal{A}) \cup \hat{\phi}_{j-1})$ is monotonically decreasing in $\mathcal{A}$, and the set $\hat{\phi}$ is shrinking in $\mathcal{A}$. □

Proof of Theorem 6.8. Suppose $\mathcal{V}, \mathcal{X}, f$ and $\mathbb{P}$ are given satisfying the requirements of Lemma A.12. Let $\mathcal{B}_1, \ldots, \mathcal{B}_s$ be the collection of all $m = \binom{n}{k}$ subsets of $\mathcal{V}$. Let $g, \mathcal{W}, \mathcal{X}'$ and $\mathbb{P}'$ denote the problem instance induced by sets $\mathcal{B}_1, \ldots, \mathcal{B}_s$, as in Lemma A.12. Note that there is a 1-1 correspondence between batch mode policies for $(f, \mathbb{P})$ and fully sequential policies for $(g, \mathbb{P}')$. Due to the adaptive submodularity of $(g, \mathbb{P}')$, and observing that by assumption $(g, \mathbb{P}')$ is self-certifying (e.g., $(g(\mathcal{S}), \mathbb{P}')$ depends only on the state of items in S) and strongly adaptively monotone (i.e., selecting more (batches of) items never hurts), it follows from Theorem 5.8 of Golovin and Krause [GK11b] that the greedy policy π (w.r.t. $(g, \mathbb{P}')$) satisfies

$$\mathrm{cost}_{avg}(\pi) \leq \mathrm{cost}_{avg}(\pi^*_{\mathrm{batch},k}) \Big(\ln Q + 1 \Big).$$

Policy π greedily assembles batches; however, each batch is chosen optimally, which is itself a combinatorially hard problem. Consider the function $\Delta_{\mathrm{B}} : 2^{\mathcal{V}} \times 2^{\mathcal{V} \times \mathcal{X}} \to \mathbb{N}$, where, for $\mathcal{B} \subseteq \mathcal{V}$ and $\mathbf{x}_{\mathcal{A}} \subseteq \mathcal{V} \times \mathcal{X}$,

$$\Delta_{\mathrm{B}}(\mathcal{B}, \mathbf{x}_{\mathcal{A}}) = \sum_{\mathbf{x}_{\mathcal{V}}} P(\mathbf{x}_{\mathcal{V}} \mid \mathbf{x}_{\mathcal{A}}) \Big[f(\mathbf{x}_{\mathcal{B}} \cup \mathbf{x}_{\mathcal{A}}) - f(\mathbf{x}_{\mathcal{A}}) \Big].$$

Note that implementing the greedy policy w.r.t. $(g, \mathbb{P}')$ requires in every step, assuming observations $\mathbf{x}_{\mathcal{A}}$ have already been made, identifying an optimal batch $i \in \mathcal{W}$ s.t.

$$\mathcal{B}_i \in \operatorname*{arg\,max}_{|\mathcal{B}| \leq k} \Delta_{\mathrm{B}}(\mathcal{B}, \mathbf{x}_{\mathcal{A}}).$$

However, it can be seen that $\Delta_{\mathrm{B}}(\mathcal{B}, \mathbf{x}_{\mathcal{A}})$ is monotonic submodular in $\mathcal{B}$ for any $\mathbf{x}_{\mathcal{A}}$, and therefore the greedy algorithm (applied to $\Delta_{\mathrm{B}}(\cdot, \mathbf{x}_{\mathcal{A}})$) produces a near-optimal batch $j \in \mathcal{W}$ such that $\Delta_g(j \mid \mathbf{x}_{\mathcal{A}}) \geq (1 - \frac{1}{e})\Delta_g(i \mid \mathbf{x}_{\mathcal{A}})$. From Theorem 2.7 we obtain

$$\mathrm{cost}_{avg}(\pi^g_{\mathrm{batch},k}) \leq \mathrm{cost}_{avg}(\pi^*_{\mathrm{batch},k}) \Big(\frac{e}{e-1}\Big) \Big(\ln Q + 1\Big).$$

Note that BatchGreedy implements exactly this greedy algorithm. The stated result about worst-case cost follows analogously from Theorem A.12 of Golovin and Krause [GK11b]. □

A.4.4 Proof of Theorem 6.9: Bounding BatchGreedy against π^*_{seq}

To begin with, we extend the definition of the conditional expected marginal benefit of an item to that of a policy:

Definition A.13. Suppose we have selected and observed items $\mathbf{x}_\mathcal{A} \subseteq \mathcal{V} \times \mathcal{X}$, the conditional expected marginal benefit of a policy π, denoted $\Delta_f(\pi \mid \mathbf{x}_\mathcal{A})$, is $\Delta_f(\pi \mid \mathbf{x}_\mathcal{A}) := \mathbb{E}\left[f(\mathbf{x}_\mathcal{A} \cup \mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) - f(\mathbf{x}_\mathcal{A}) \mid \mathbf{x}_\mathcal{A}\right]$, where the expectation is computed w.r.t. $P(\mathbf{X}_\mathcal{V} \mid \mathbf{x}_\mathcal{A})$.

We now state a Lemma which slightly generalizes Lemma A.9 of Golovin and Krause [GK11b].

Lemma A.14. *Assume each item has a unit cost. Let π^* be any policy, which only picks a number of items divisible by k, and $\mathbf{x}_\mathcal{A} \subseteq \mathcal{V} \times \mathcal{X}$.*

$$\Delta_f(\pi^* \mid \mathbf{x}_\mathcal{A}) \leq \frac{\text{cost}_{avg}(\pi^* \mid \mathbf{x}_\mathcal{A})}{k} \max_{\pi : \text{cost}_{avg}(\pi) \leq k} \Delta_f(\pi \mid \mathbf{x}_\mathcal{A})$$

Proof. Let π be the policy that attempts to select $\mathbf{x}_\mathcal{A}$, terminating if observing inconsistent observations, and then executes π^*. Now consider any policy of cost exactly k, and let $w(\pi')$ denote the probability that the subtree π' is contained in π. By adaptive submodularity, it holds that the total contribution of π' to $\Delta_f(\pi^* \mid \mathbf{x}_\mathcal{A})$ is bounded by $w(\pi')\Delta_f(\pi' \mid \mathbf{x}_\mathcal{A})$, and therefore

$$\Delta_f(\pi^* \mid \mathbf{x}_\mathcal{A}) \leq \sum_{\pi' : \text{cost}_{avg}(\pi') \leq k} w(\pi')\Delta_f(\pi' \mid \mathbf{x}_\mathcal{A}).$$

Note that each policy π' contributes cost $w(\pi')k$ to $\text{cost}_{avg}(\pi^* \mid \mathbf{x}_\mathcal{A})$. Therefore,

$$\sum_{\pi' : \text{cost}_{avg}(\pi') \leq k} w(\pi')k \leq \text{cost}_{avg}(\pi^* \mid \mathbf{x}_\mathcal{A}).$$

Therefore,

$$\Delta_f(\pi^* \mid \mathbf{x}_\mathcal{A}) \leq \sum_{\pi' : \text{cost}_{avg}(\pi') \leq k} w(\pi')\Delta_f(\pi' \mid \mathbf{x}_\mathcal{A}) \leq \text{cost}_{avg}(\pi^* \mid \mathbf{x}_\mathcal{A}) \max_{\pi''} \frac{\Delta_f(\pi'' \mid \mathbf{x}_\mathcal{A})}{k}.$$

□

Theorem A.15. *Fix any $\alpha \geq 1$ and let $\gamma = \textsc{gap}_k \cdot \frac{e}{e-1}$. If f is adaptive monotone and adaptive submodular on $\mathbb{P}[\mathbf{X}_\mathcal{V}]$, and π is the* BatchGreedy *policy, then for all policies π^* selecting a number of items divisible by k, and positive integers ℓ and m*

$$F(\pi_{[\ell k]}) > \left(1 - e^{-\ell/\alpha\gamma m}\right) F(\pi^*_{[mk]}).$$

where $\pi_{[i]}$ denotes the level-i-truncation of π obtained by running until it terminates or until it selects i items.

Proof. The proof goes along the lines of the performance analysis of the greedy algorithm for maximizing a submodular function subject to a cardinality constraint of Nemhauser, Wolsey, and Fisher [NWF78], and its extension to the adaptive setting by Golovin and Krause [GK11b].

We consider breaking the optimal policy into phases of length k. Without loss of generality we assume $\pi^* = \pi^*_{[mk]}$. We derive a sequence of inequalities:

$$F(\pi^*) \quad \leq \quad F(\pi_{[ik]}) + \alpha\gamma m \left(F(\pi_{[(i+1)k]}) - F(\pi_{[ik]})\right). \tag{A.4.4}$$

These inequalities follow from adaptive monotonicity and Lemma A.14. Here, the factor $\textsc{gap}_k$ in γ is because we are approximating the optimal sequential policy of length k with a non-adaptive policy. Applying the greedy algorithm to construct the nonadaptive policy contributes a further factor $e/(e-1)$ to γ.

Using an argument as in the proof of Theorem A.10 from Golovin and Krause [GK11b], we then have

$$F(\pi_{[(i+1)k]}) - F(\pi_{[ik]}) \quad \geq \frac{F(\pi_{[ik]}@\pi^*) - F(\pi_{[ik]})}{\alpha\gamma m}$$

where @ denotes policy concatenation [2].

Now define $\Delta_i := F(\pi^*) - F(\pi_{[ik]})$, so that (A.4.4) implies $\Delta_i \leq \alpha\gamma m(\Delta_i - \Delta_{i+1})$, from which we infer $\Delta_{i+1} \leq \left(1 - \frac{1}{\alpha\gamma m}\right)\Delta_i$ and hence $\Delta_\ell \leq \left(1 - \frac{1}{\alpha\gamma m}\right)^\ell \Delta_0 < e^{-\ell/\alpha\gamma m}\Delta_0$, where for this last inequality we have used the fact that $1 - x < e^{-x}$ for all $x > 0$. Thus $F(\pi^*) - F(\pi_{[\ell k]}) < e^{-\ell/\alpha\gamma m}\left(F(\pi^*) - F(\pi_{[0]})\right) \leq e^{-\ell/\alpha\gamma m}F(\pi^*)$ so $F(\pi) > (1 - e^{-\ell/\alpha\gamma m})F(\pi^*)$. □

[2] According to Definition A.6 in Golovin and Krause [GK11b], the concatenation of $\pi_1@\pi_2$ is defined as the policy obtained by running policy π_1 to completion, and then running policy π_2 as if from a fresh start, ignoring the information gathered during the running of π_1.

Proof of Theorem 6.9. Theorem 6.9 follows as an immediate corollary to Theorem A.15.

Let $\beta > 0$. Let m be the smallest number, so that there exists a fully sequential policy π^* of length mk with value $F(\pi^*) \geq Q$. Then running BatchGreedy $\pi^g_{\text{batch},k}$ for ℓ batches of size k, where

$$\ell = \lceil \gamma \ln Q/\beta \rceil m,$$

is sufficient so that $F(\pi^g_{\text{batch},k}) \geq Q - \beta$. Now suppose $P(f(\mathcal{S}(\pi^g_{\text{batch},k}, \mathbf{x}_\mathcal{V})) \leq Q - 1) > \beta$. Then

$$F(\pi^g_{\text{batch},k}) < \beta(Q-1) + (1-\beta)Q = Q - \beta,$$

a contradiction. □

A.5 Proofs from Chapter 7

A.5.1 Proof of Theorem 7.3: Upper Bound on the Regret of OnlineVoI

We now proceed to prove the bound on the expected regret of our online learning algorithm.

Proof of Theorem 7.3. One way to model the non-myopic value of information problem is to view it as a (finite horizon) Partially Observable Markov Decision Process (POMDP), where each (belief-) state represents the selected tests and observed outcome of each test. Formally, the POMDP can be written as

$$M \triangleq \left(\Omega, \mathcal{V}, R^M, P^M, \tau, \rho\right). \tag{A.5.1}$$

Here, Ω is the set of belief states, $\mathcal{V}$ is the set of actions (i.e., tests), $R_v^M(b)$ is the (expected) reward associated with action v while in belief state b, $P_v^M(b' \mid b)$ denotes the probability of transitioning to state b' if action v is selected while in state b, τ is the time horizon for each session, and ρ is the initial belief state distribution.

In our problem, the *transition probabilities* P^M can be fully specified by the conditional probabilities of the test outcomes given the hidden state $\mathbb{P}[x_v \mid \theta]$; the *prior* distribution ρ on belief states can be specified by the prior distribution on the hypotheses $\mathbb{P}[\theta]$, and $\mathbb{P}[x_v \mid \theta]$. The *reward* R^M for running a policy π on M is the utility achieved upon termination of the policy. More specifically, we can interpret the reward function R^M as follows: we get reward 0 as the policy keeps selecting new tests, but get (expected) reward $\mathrm{VoI}(\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})) \triangleq \max_{y \in \mathcal{Y}} \mathbb{E}_\theta[u(\theta, y) \mid \mathcal{S}(\pi, \mathbf{x}_\mathcal{V})]$ if the policy terminates upon observing $\mathcal{S}(\pi, \mathbf{x}_\mathcal{V})$ and suggests a decision. The reward function measures the expected (total) utility one can get by making a decision after running policy π.

We now consider running Algorithm 11 over k sessions of fixed duration τ. Following the previous discussion, the problem is equivalent to learning to optimize a random finite horizon POMDP of length τ in k repeated episodes of interaction. To establish the regret bound of Theorem 7.3, we need the following result:

Theorem A.16 (Theorem 1 of Osband, Russo, and Van Roy [ORVR13])**.** *Consider the problem of learning to optimize a random finite horizon (PO)MDP* $M = \left(\Omega, \mathcal{V}, R^M, P^M, \tau, \rho\right)$

in k repeated episodes, and consider running the following algorithm: at the start of each episode it updates the prior distribution over the MDP and takes one sample from the posterior, and then follows the policy that is optimal for this sampled MDP. For any prior distribution on the MDPs, it holds that

$$\mathbb{E}[Regret(k,\tau)] = O\left(\tau|\Omega|\sqrt{k\tau|\mathcal{V}|\log(k\tau|\Omega||\mathcal{V}|)}\right).$$

Theorem A.16 implies that the posterior sampling strategy as employed in Algorithm 11 allows efficient learning of the MDP, given that one can find the *optimal* policy for the sampled MDP at each episode. However, since finding the optimal policy is NP-hard, in practice we can only *approximate* the optimal policy. In Algorithm 11, we consider running the greedy policy (i.e., Algorithm 3) in each epoch (i.e., episode) to solve the sampled MDP:

Corollary A.17. *Let M be a sampled MDP, and c^*_{wc} be the worst-case cost of the (worst-case) optimal policy on M. Consider running Algorithm 3 for $\tau = (2\ln(1/\delta)+1)\, c^*_{wc}$ steps. Then, with probability at least $1-\eta$, it achieves the optimal VoI on M.*

Proof of Corollary A.17. By Theorem 4.15, we know that the greedy policy finds the target decision region with probability at least $1-\eta$. Furthermore, by definition we know that each decision region $\mathcal{R}_y = \{h : h \equiv \mathbf{x}_\mathcal{V} \wedge U(y \mid \mathbf{x}_\mathcal{V}) = \text{VoI}(\mathbf{x}_\mathcal{V})\}$ represents an optimal action for any of its enclosed hypotheses. In other words, a policy that successfully outputs a decision region achieves the optimal VoI. □

Recall from §7.3.2 that we denote the optimal policy on the sampled MDP in epoch ℓ as $\pi^*_{\tilde{P}_\ell}$. From Corollary A.17, we know that Algorithm 3 achieves optimal utility with probability at least $1-\eta$. Hence, the expected "regret" of Algorithm 3 over $\pi^*_{\tilde{P}_\ell}$ is

$$\tilde{\Delta}_\ell \overset{(a)}{\leq} (1-\eta)\cdot 0 + \eta \cdot 1 = \eta. \tag{A.5.2}$$

Here, step (a) is because the utility is normalized so that $U \in [0,1]$. Note that $\tilde{\Delta}_\ell$ in Equation (A.5.2) refers to the difference between the value of Algorithm 3 and the value of the optimal policy on the sampled MDP (not the optimal policy for the true MDP). In other words, the price of not following the optimal policy is at most η.

By Theorem A.16, we know that following OPT_i for epoch i achieves expected regret

$$O\left(\tau|\Omega|\sqrt{k\tau|\mathcal{V}|\log(k\tau|\Omega||\mathcal{V}|)}\right).$$

Further, we know that the price of approximating the optimal policy at epoch i is at most η. Combining these two results we get

$$\begin{aligned}\mathbb{E}[\text{Regret}(k,\tau)] &= O\left(\tau|\Omega|\sqrt{k\tau|\mathcal{V}|\log(k\tau|\Omega||\mathcal{V}|)}\right) + \sum_{i=1}^{k}\eta \\ &= O\left(\tau|\Omega|\sqrt{k\tau|\mathcal{V}|\log(k\tau|\Omega||\mathcal{V}|)} + \eta k\right),\end{aligned}$$

where $|\Omega| = S$ represents the number of the belief states, $|\mathcal{V}| = t$ represents the number of tests. Hence it completes the proof. □

Bibliography

[AD92] A. C. Atkinson and A. N. Donev. *Optimum Experiment Designs*. Oxford UP, 1992 (cit. on pp. 4, 21).

[AF04] Yotam Abramson and Yoav Freund. *Active learning for visual object recognition*. Tech. rep. UCSD, 2004 (cit. on p. 137).

[AG12] Shipra Agrawal and Navin Goyal. "Analysis of Thompson Sampling for the Multi-armed Bandit Problem". In: *The 25th Annual Conference on Learning Theory (COLT)*. 2012 (cit. on p. 165).

[Agr+12] D. Agrawal, P. Bernstein, E. Bertino, S. Davidson, U. Dayal, Franklin M., and J. Widom. *Challenges and Opportunities with Big Data: A white paper prepared for the Computing Community Consortium committee of the Computing Research Association*. Tech. rep. 2012 (cit. on p. 3).

[AN15] Arash Asadpour and Hamid Nazerzadeh. "Maximizing stochastic monotone submodular functions". In: *Management Science* (2015) (cit. on pp. 42, 146, 147, 240).

[Ans+09] Elliot Anshelevich, Deeparnab Chakrabarty, Ameya Hate, and Chaitanya Swamy. "Approximation Algorithms for the Firefighter Problem: Cuts over Time and Submodularity". In: *Algorithms and Computation*. Springer Berlin / Heidelberg, 2009 (cit. on p. 15).

[Ark+93] Esther M. Arkin, Henk Meijer, Joseph S. B. Mitchell, David Rappaport, and Steven S. Skiena. "Decision trees for geometric models". In: *SCG '93: Proceedings of the ninth annual Symposium on Computational Geometry*. San Diego, California, United States: ACM, 1993, pp. 369–378 (cit. on p. 78).

[Bal81] Dana H Ballard. "Generalizing the Hough transform to detect arbitrary shapes". In: *Pattern recognition* (1981) (cit. on p. 119).

[BBB13] Mohamed-Rafik Bouguelia, Yolande Belaïd, and Abdel Belaïd. "A stream-based semi-supervised active learning approach for document classification". In: *2013 12th International Conference on Document Analysis and Recognition*. IEEE. 2013, pp. 611–615 (cit. on p. 35).

[BBL06] M. Balcan, A. Beygelzimer, and J. Langford. "Agnostic active learning". In: *ICML*. 2006 (cit. on pp. 4, 21, 37, 39, 40).

[BBZ07] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. "Margin based active learning". In: *Learning Theory*. Springer, 2007, pp. 35–50 (cit. on p. 37).

[Ber+10] Scott M Berry, Bradley P Carlin, J Jack Lee, and Peter Muller. *Bayesian adaptive methods for clinical trials*. CRC press, 2010 (cit. on p. 7).

[BG07] Mustafa Bilgic and Lise Getoor. "VOILA: Efficient Feature-value Acquisition for Classification". In: *Twenty-Second Conference on Artificial Intelligence (AAAI)*. 2007 (cit. on p. 4).

[BHe06] J. Berry, W. Hart, and et.al. "A Facility Location Approach to Sensor Placement Optimization". In: *8th Annual Symposium on Water Distribution Systems Analysis*. Cincinnati, Ohio, 2006 (cit. on p. 4).

[BHZ10] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Morteza Zadimoghaddam. "Submodular secretary problem and extensions". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2010, pp. 39–52 (cit. on p. 35).

[Bie12] Alberto Bietti. *Active Learning for Object Detection on Satellite Images*. Tech. rep. Caltech, 2012 (cit. on p. 137).

[BK03] Greg Barish and Craig A. Knoblock. "Learning Value Predictors for the Speculative Execution of Information Gathering Plans". In: *IJCAI*. 2003 (cit. on p. 4).

[BL13a] Maria-Florina Balcan and Philip M Long. "Active and passive learning of linear separators under log-concave distributions." In: *COLT*. 2013, pp. 288–316 (cit. on p. 39).

[BL13b] Sebastien Bubeck and Che-Yu Liu. "Prior-free and prior-dependent regret bounds for Thompson Sampling". In: *Proc. Neural Information Processing Systems (NIPS)*. 2013 (cit. on p. 165).

[BLK12a] O. Barinova, V. Lempitsky, and P. Kholi. "On Detection of Multiple Object Instances Using Hough Transforms". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.9 (2012), pp. 1773–1784. DOI: 10.1109/tpami.2012.79 (cit. on pp. 132, 135, 137).

[BLK12b] Olga Barinova, Victor Lempitsky, and Pushmeet Kholi. "On detection of multiple object instances using hough transforms". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.9 (2012), pp. 1773–1784 (cit. on p. 26).

[BT52] Ralph Allan Bradley and Milton E Terry. "Rank analysis of incomplete block designs: I. The method of paired comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345 (cit. on p. 106).

[BU15] Maria-Florina Balcan and Ruth Urner. "Active Learning–Modern Learning Theory". In: (2015) (cit. on p. 39).

[Buc+12] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. "A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization". In: *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*. IEEE. 2012, pp. 649–658 (cit. on p. 33).

[Buc+14] Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. "Submodular maximization with cardinality constraints". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. 2014, pp. 1433–1452 (cit. on p. 33).

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004 (cit. on pp. 21, 24, 33).

[BWHS05] Tanya Y. Berger-Wolf, William E. Hart, and Jared Saia. "Discrete Sensor Placement Problems in Distribution Networks". In: *Journal of Mathematical and Computer Modelling* 42.13 (2005), pp. 1385–1396 (cit. on p. 4).

[Cal+07] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. "Maximizing a submodular set function subject to a matroid constraint". In: *International Conference on Integer Programming and Combinatorial Optimization*. Springer. 2007, pp. 182–196 (cit. on p. 33).

[CF09] Carri W Chan and Vivek F Farias. "Stochastic depletion problems: Effective myopic policies for a class of dynamic optimization problems". In: *Mathematics of Operations Research* 34.2 (2009), pp. 333–350 (cit. on p. 33).

[CG07] Anton Chechetka and Carlos Guestrin. "Efficient Principled Learning of Thin Junction Trees". In: *In Advances in Neural Information Processing Systems (NIPS)*. Vancouver, Canada, 2007 (cit. on p. 28).

[CGJ96] D.A. Cohn, Z. Ghahramani, and M.I. Jordan. "Active Learning with Statistical Models". In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 129–145 (cit. on p. 4).

[Cha+07] Venkatesan T Chakaravarthy, Vinayaka Pandit, Sambuddha Roy, Pranjal Awasthi, and Mukesh Mohania. "Decision trees for entity identification: Approximation algorithms and hardness results". In: *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM. 2007, pp. 53–62 (cit. on p. 40).

[Che+14] Yuxin Chen, Hiroaki Shioi, Cesar Fuentes Montesinos, Lian Pin Koh, Serge Wich, and Andreas Krause. "Active Detection via Adaptive Submodularity". In: *Proc. International Conference on Machine Learning (ICML)*. 2014 (cit. on p. 18).

[Che+15a] Yuxin Chen, S. Hamed Hassani, Amin Karbasi, and Andreas Krause. "Sequential Information Maximization: When is Greedy Near-optimal?" In: *Proc. International Conference on Learning Theory (COLT)*. 2015 (cit. on p. 18).

[Che+15b] Yuxin Chen, Shervin Javdani, Amin Karbasi, James Andrew Bagnell, Siddhartha Srinivasa, and Andreas Krause. "Submodular Surrogates for Value of Information". In: *Proc. Conference on Artificial Intelligence (AAAI)*. 2015 (cit. on p. 18).

[Che+16] Yuxin Chen, Jean-Michel Renders, Morteza H. Chehreghani, and Andreas Krause. *Efficient Online Learning for Optimizing Value of Information: Theory and Application to Interactive Troubleshooting*. Tech. rep. 2016 (cit. on p. 18).

[Che81] Herman Chernoff. "A note on an inequality involving the normal distribution". In: *The Annals of Probability* (1981), pp. 533–535 (cit. on p. 36).

[CHK17] Yuxin Chen, S. Hamed Hassani, and Andreas Krause. "Near-optimal Bayesian Active Learning with Correlated and Noisy Tests". In: *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2017 (cit. on p. 18).

[Chu+11] Wei Chu, Martin Zinkevich, Lihong Li, Achint Thomas, and Belle Tseng. "Unbiased online active learning in data streams". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 195–203 (cit. on p. 35).

[CK11] Imre Csiszar and János Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011 (cit. on p. 184).

[CK13] Yuxin Chen and Andreas Krause. "Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization". In: *International Conference on Machine Learning (ICML)*. 2013 (cit. on p. 18).

[CL06] Nicolo Cesa–Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006 (cit. on p. 37).

[CL11] Olivier Chapelle and Lihong Li. "An Empirical Evaluation of Thompson Sampling". In: *Proc. Neural Information Processing Systems (NIPS)*. 2011 (cit. on p. 165).

[CM02] Dorin Comaniciu and Peter Meer. "Mean shift: A robust approach toward feature space analysis". In: *PAMI*. 2002 (cit. on p. 129).

[Coh94] David A Cohn. "Neural network exploration using optimal experiment design". In: *Advances in Neural Information Processing Systems*. 1994, 679—686 (cit. on p. 36).

[CT12] Maya Cakmak and Andrea L Thomaz. "Designing robot learners that ask good questions". In: *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM. 2012, pp. 17–24 (cit. on p. 38).

[CT14] Sonia Chernova and Andrea L Thomaz. "Robot learning from human teachers". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8.3 (2014), pp. 1–121 (cit. on p. 38).

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Interscience, 1991 (cit. on p. 25).

[Cur+88] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. *A Bayesian Approach to the Design and Analysis of Computer Experiments*. Tech. rep. ORNL-6498. Oak Ridge National Laboratory, Sept. 1988 (cit. on pp. 4, 21).

[CV95] K. Chaloner and I. Verdinelli. "Bayesian Experimental Design: A Review". In: *Statistical Science* 10.3 (1995), pp. 273–304 (cit. on pp. 4, 21).

[CWN05] R. Castro, R. Willett, and R. Nowak. "Faster Rates in Regression Via Active Learning". In: *NIPS*. 2005 (cit. on pp. 4, 39).

[DA88] A. N. Donev and A. C. Atkinson. "Construction of Exact D-Optimum Experimental Designs". In: *Technometrics* 30.4 (1988), pp. 429–433 (cit. on p. 21).

[Das05a] S. Dasgupta. "Coarse sample complexity bounds for active learning". In: *NIPS*. 2005 (cit. on pp. 21, 39).

[Das05b] Sanjoy Dasgupta. "Analysis of a greedy active learning strategy". In: *Advances in Neural Information Processing Systems 17*. 2005 (cit. on pp. 9, 31, 40).

[DHK14] A. Deshpande, L. Hellerstein, and D. Kletenik. "Approximation Algorithms for Stochastic Boolean Function Evaluation and Stochastic Submodular Set Cover". In: *SODA*. 2014 (cit. on pp. 33, 41, 85).

[DHM07] S. Dasgupta, D.J. Hsu, and C. Monteleoni. "A general agnostic active learning algorithm". In: *NIPS*. 2007 (cit. on pp. 4, 37).

[DJ97] S. Dittmer and F. Jensen. "Myopic Value of Information in Influence Diagrams". In: *UAI*. San Francisco, 1997, pp. 142–149 (cit. on pp. 4, 27).

[DK08] A. Das and D. Kempe. "Algorithms for Subset Selection in Linear Regression". In: *Proceedings of the ACM Symposium on Theory of Computing*. 2008, pp. 45–54 (cit. on p. 28).

[DKM05] S. Dasgupta, A. Kalai, and C. Monteleoni. "Analysis of perceptron-based active learning". In: *Eighteenth Annual Conference on Learning Theory*. 2005 (cit. on p. 26).

[DL11] Sanjoy Dasgupta and J Langford. "Active learning". In: *Encyclopedia of Machine Learning* (2011) (cit. on p. 6).

[DMH07] Sanjoy Dasgupta, Claire Monteleoni, and Daniel J Hsu. "A general agnostic active learning algorithm". In: *Advances in neural information processing systems*. 2007, pp. 353–360 (cit. on p. 39).

[DSM09] Gregory Druck, Burr Settles, and Andrew McCallum. "Active learning by labeling features". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics. 2009, pp. 81–90 (cit. on p. 38).

[DT05] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *CVPR*. 2005 (cit. on p. 136).

[FBT98] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. "Active Markov Localization for Mobile Robots". In: *Robotics and Autonomous Systems* 25 (1998), pp. 195–207 (cit. on p. 7).

[Fel+10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 32.9 (Sept. 2010), pp. 1627–1645 (cit. on pp. 135–137).

[Fer89] Thomas S Ferguson. "Who solved the secretary problem?" In: *Statistical science* (1989), pp. 282–289 (cit. on p. 35).

[Fre+97] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. "Selective sampling using the query by committee algorithm". In: *Machine learning* 28.2-3 (1997), pp. 133–168 (cit. on p. 40).

[Fuj00] Toshihiro Fujito. "Approximation algorithms for submodular set cover with applications". In: *IEICE Transactions on Information and Systems* 83.3 (2000), pp. 480–487 (cit. on p. 32).

[Gar+12] Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard Mann. "Bayesian Optimal Active Search and Surveying". In: *Proceredings of the 29th Annual International Conference on Machine Learning (ICML)*. 2012 (cit. on p. 37).

[GB10] Andrew Guillory and Jeff A Bilmes. "Interactive Submodular Set Cover". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 415–422 (cit. on p. 32).

[GB11a] Andrew Guillory and Jeff Bilmes. "Active Semi-Supervised Learning using Submodular Functions". In: *UAI*. 2011 (cit. on pp. 4, 27, 28, 37).

[GB11b] Andrew Guillory and Jeff Bilmes. "Simultaneous Learning and Covering with Adversarial Noise". In: *ICML*. 2011, pp. 369–376 (cit. on p. 85).

[GBL01] H. H. Gonzalez-Banos and J. Latombe. "A Randomized Art-Gallery Algorithm for Sensor Placement". In: *Proc. 17th ACM Symposium on Computational Geometry*. 2001, pp. 232–240 (cit. on p. 4).

[GFM11] Ross Girshick, Pedro Felzenszwalb, and David McAllester. "Object detection with grammar models". In: *IEEE TPAMI* (2011) (cit. on p. 135).

[GG74] M. R. Garey and Ronald L. Graham. "Performance Bounds on the Splitting Algorithm for Binary Testing". In: *Acta Inf.* 3 (1974), pp. 347–355 (cit. on p. 78).

[GG84] S. Geman and D. Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6.6 (1984), pp. 721–741 (cit. on p. 22).

[GJ79] J. C. Gittins and D. M. Jones. "A Dynamic Allocation Index for the Discounted Multiarmed Bandit Problem". In: *Biometrika* 66.3 (1979), pp. 561–565 (cit. on p. 4).

[GK11a] Daniel Golovin and Andreas Krause. "Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization". In: *JAIR* (2011), pp. 427–486 (cit. on pp. 9, 24, 53, 78, 80, 84, 142, 198).

[GK11b] Daniel Golovin and Andreas Krause. "Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization". In: *Journal of Artificial Intelligence Research (JAIR)* 42 (2011), pp. 427–486 (cit. on pp. 15, 26–28, 31, 78, 125, 126, 150, 152, 153, 173, 201, 204, 244–247).

[GKK15] Alkis Gotovos, Amin Karbasi, and Andreas Krause. "Non-monotone Adaptive Submodular Maximization". In: *International Joint Conference on Artificial Intelligence (IJCAI)*. 2015 (cit. on p. 33).

[GKR10a] D. Golovin, A. Krause, and D. Ray. "Near-Optimal Bayesian Active Learning with Noisy Observations". In: *CoRR* (2010) (cit. on pp. 11, 68, 74, 75, 78, 82, 195, 197).

[GKR10b] Daniel Golovin, Andreas Krause, and Debajyoti Ray. "Near-Optimal Bayesian Active Learning with Noisy Observations". In: *Proc. Neural Information Processing Systems (NIPS)*. 2010 (cit. on p. 232).

[GL09] Juergen Gall and Victor S. Lempitsky. "Class-specific Hough forests for object detection". In: *CVPR*. 2009, pp. 1022–1029 (cit. on p. 130).

[GLM01] Jacob Goldenberg, Barak Libai, and Eitan Muller. "Talk of the network: A complex systems look at the underlying process of word-of-mouth". In: *Marketing letters* 12.3 (2001), pp. 211–223 (cit. on p. 142).

[GLS08] Navin Goyal, Yury Lifshits, and Hinrich Schütze. "Disorder Inequality: A Combinatorial Approach to Nearest Neighbor Search". In: *WSDM*. 2008 (cit. on p. 7).

[GLS81] M. Grotschel, L. Lovász, and A. Schrijver. "The ellipsoid method and its consequences in combinatorial optimization". In: *Combinatorica* 1 (1981), pp. 169–197 (cit. on p. 28).

[Gol+11] Daniel Golovin, Andreas Krause, Beth Gardner, Sarah Converse, and Steve Morey. "Dynamic Resource Allocation in Conservation Planning". In: *AAAI*. 2011 (cit. on p. 15).

[Gra+10] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich. "Web-Scale Bayesian Click-Through rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine." In: *ICML*. 2010, pp. 13–20 (cit. on p. 165).

[Gra78] Mark Granovetter. "Threshold models of collective behavior". In: *American journal of sociology* (1978), pp. 1420–1443 (cit. on p. 142).

[GS07] Pranava R. Goundan and Andreas S. Schulz. *Revisiting the Greedy Approach to Submodular Set Function Maximization*. Tech. rep. Massachusetts Institute of Technology, 2007 (cit. on p. 241).

[GS88] Rodney M Goodman and Padhraic Smyth. "Decision tree design from a communication theory standpoint". In: *IEEE Transactions on Information Theory* 34.5 (1988), pp. 979–994 (cit. on p. 41).

[GSSS11] Alon Gonen, Sivan Sabato, and Shai Shalev-Shwartz. "Active Learning Halfspaces under Margin Assumptions". In: *CoRR* abs/1112.1556v3 (2011) (cit. on p. 153).

[GSSS13] Alon Gonen, Sivan Sabato, and Shai Shalev-Shwartz. "Efficient active learning of halfspaces: an aggressive approach". In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 2583–2615 (cit. on p. 39).

[Gup+10] Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, and R. Ravi. "Approximation Algorithms for Optimal Decision Trees and Adaptive TSP Problems". In: *To appear in Proc. ICALP*. 1003.0722. 2010 (cit. on p. 78).

[Han+11] Steve Hanneke et al. "Rates of convergence in active learning". In: *The Annals of Statistics* 39.1 (2011), pp. 333–361 (cit. on p. 39).

[Han07a] Steve Hanneke. "A bound on the label complexity of agnostic active learning". In: *ICML*. 2007 (cit. on p. 37).

[Han07b] Steve Hanneke. "Teaching dimension and the complexity of active learning". In: *Learning Theory*. Springer, 2007, pp. 66–81 (cit. on p. 37).

[Hau+06] Alexander G Hauptmann, Wei-Hao Lin, Rong Yan, Jun Yang, and Ming-Yu Chen. "Extreme video retrieval: joint maximization of human and computer performance". In: *Proceedings of the 14th ACM international conference on Multimedia*. ACM. 2006, pp. 385–394 (cit. on p. 35).

[Hau+12] Jarvis Haupt, Richard G Baraniuk, Rui M Castro, and Robert D Nowak. "Sequentially designed compressed sensing." In: *SSP*. 2012, pp. 401–404 (cit. on p. 26).

[HBR94] David Heckerman, John Breese, and Koos Rommelse. *Troubleshooting under uncertainty*. Tech. rep. Technical Report MSR-TR-94-07, Microsoft Research, 1994 (cit. on p. 7).

[Her+99] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. "An Algorithmic Framework for Performing Collaborative Filtering". In: *SIGIR*. 1999 (cit. on p. 107).

[HHM93] David Heckerman, E. Horvitz, and B. Middleton. "An Approximate Nonmyopic Computation for Value of Information". In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 15 (1993), pp. 292–298 (cit. on pp. 4, 21).

[HM84] R. A. Howard and J.E. Matheson. "Readings on the Principles and Applications of Decision Analysis II". In: Reprinted 2005 in Decision Analysis 2(3) 127-143. Strategic Decision Group, Menlo Park, 1984. Chap. Influence Diagrams, pp. 719–762 (cit. on p. 4).

[Hoi+06a] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. "Batch mode active learning and its application to medical image classification". In: *Proceedings of the 23rd international conference on Machine learning*. ICML '06. Pittsburgh, Pennsylvania: ACM, 2006, pp. 417–424 (cit. on pp. 27, 37).

[Hoi+06b] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. "Batch mode active learning and its application to medical image classification". In: *ICML*. Pittsburgh, Pennsylvania, 2006 (cit. on pp. 156, 157).

[Hor63] Michael Horstein. "Sequential transmission using noiseless feedback". In: *Information Theory, IEEE Transactions on* 9.3 (1963), pp. 136–143 (cit. on p. 41).

[Hou59] P. V. C. Hough. "Machine Analysis of Bubble Chamber Pictures". In: *International Conference on High Energy Accelerators and Instrumentation*. 1959 (cit. on p. 119).

[How66] R. A. Howard. "Information value theory". In: *IEEE Transactions on Systems Science and Cybernetics*. 1966 (cit. on pp. 4, 21, 25).

[HY14] Steve Hanneke and Liu Yang. "Minimax Analysis of Active Learning". In: *CoRR* abs/1410.0996 (2014) (cit. on pp. 37, 39).

[HY15] Bryan D He and Yisong Yue. "Smooth Interactive Submodular Set Cover". In: *Advances in Neural Information Processing Systems*. 2015, pp. 118–126 (cit. on p. 32).

[Jav+13] Shervin Javdani, Matthew Klingensmith, J. Andrew (Drew) Bagnell, Nancy Pollard, and Siddhartha Srinivasa. "Efficient Touch Based Localization through Submodularity". In: *IEEE ICRA*. 2013 (cit. on p. 11).

[Jav+14] S. Javdani, Y. Chen, A. Karbasi, A. Krause, D. Bagnell, and S. Srinivasa. "Near-Optimal Bayesian Active Learning for Decision Making". In: *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2014 (cit. on pp. 18, 80).

[Joa+09] Thorsten Joachims, Thomas Hofmann, Yisong Yue, and Chun-Nam Yu. "Predicting structured objects with support vector machines". In: *Communications of the ACM* 52.11 (2009), pp. 97–104 (cit. on pp. 38, 179).

[JPC07] Shihao Ji, Ronald Parr, and Lawrence Carin. "Non-Myopic Multi-Aspect Sensing with Partially Observable Markov Decision Processes". In: *IEEE Transactions on Signal Processing* 55.6 (2007), pp. 2720–2730 (cit. on p. 4).

[JPP12] Ajay J Joshi, Fatih Porikli, and Nikolaos P Papanikolopoulos. "Scalable Active Learning for Multiclass Image Classification". In: *PAMI* (2012) (cit. on p. 137).

[JVG10] Prateek Jain, Sudheendra Vijayanarasimhan, and Kristen Grauman. "Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning". In: *Advances in Neural Information Processing Systems 23*. 2010, pp. 928–936 (cit. on p. 156).

[Kap+07] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. "Active learning with gaussian processes for object categorization". In: *ICCV*. 2007 (cit. on p. 137).

[KB03] Koushik Kar and Suman Banerjee. "Node Placement for Connected Coverage in Sensor Networks". In: *WiOpt*. 2003 (cit. on p. 4).

[KF08] D. Koller and N. Friedman. *Structured Probabilistic Models*. Electronic Preprint, 2008 (cit. on p. 21).

[KG05] Andreas Krause and Carlos Guestrin. "Near-optimal Nonmyopic Value of Information in Graphical Models". In: *Conference on Uncertainty in Artificial Intelligence (UAI)*. 2005 (cit. on p. 8).

[KG07] Andreas Krause and Carlos Guestrin. "Near-optimal Observation Selection using Submodular Functions". In: *AAAI Nectar track*. 2007 (cit. on p. 28).

[KG09] Andreas Krause and Carlos Guestrin. "Optimal Value of Information in Graphical Models". In: *JAIR* 35 (2009), pp. 557–591 (cit. on p. 28).

[KHB07] Ashish Kapoor, Eric Horvitz, and Sumit Basu. "Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning". In: *International Joint Conference on Artificial Intelligence (IJCAI)*. 2007 (cit. on p. 27).

[KIM12] Amin Karbasi, Stratis Ioannidis, and Laurent Massoulie. "Comparison-Based Learning with Rank Nets". In: *ICML*. 2012 (cit. on pp. 11, 12).

[KK07] Richard M Karp and Robert Kleinberg. "Noisy binary search and its applications". In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2007, pp. 881–890 (cit. on p. 40).

[KKM05] Haim Kaplan, Eyal Kushilevitz, and Yishay Mansour. "Learning with attribute costs". In: *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*. ACM. 2005, pp. 356–365 (cit. on p. 35).

[KKM12] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. "Thompson Sampling: An Asymptotically Optimal Finite-time Analysis". In: *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*. ALT'12. 2012, pp. 199–213 (cit. on p. 165).

[KKT03] David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network". In: *KDD*. Washington, D.C., 2003, pp. 137–146 (cit. on pp. 26, 142, 152, 154).

[KLQ95] Chun-Wa Ko, Jon Lee, and Maurice Queyranne. "An exact algorithm for maximum entropy sampling". In: *Operations Research* 43.4 (1995), pp. 684–691 (cit. on p. 8).

[Kon01] Igor Kononenko. "Machine learning for medical diagnosis: history, state of the art and perspective". In: *Artificial Intelligence in Medicine* 23 (2001), pp. 89–109 (cit. on p. 7).

[KPB99a] S Rao Kosaraju, Teresa M Przytycka, and Ryan Borgstrom. "On an optimal split tree problem". In: *Algorithms and Data Structures*. Springer, 1999, pp. 157–168 (cit. on pp. 31, 40).

[KPB99b] S. Rao Kosaraju, Teresa M. Przytycka, and Ryan S. Borgstrom. "On an Optimal Split Tree Problem". In: *WADS '99: Proceedings of the 6th International Workshop on Algorithms and Data Structures*. London, UK: Springer-Verlag, 1999, pp. 157–168 (cit. on pp. 78, 168).

[KR08] Thomas Kollar and Nicholas Roy. "Efficient Optimization of Information-Theoretic Exploration in SLAM". In: *AAAI*. 2008 (cit. on p. 4).

[Kra+06] Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. "Near-optimal sensor placements: Maximizing information while minimizing communication cost". In: *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM. 2006, pp. 2–10 (cit. on p. 41).

[Kra+09] A. Krause, B. McMahan, C. Guestrin, and A. Gupta. "Robust Submodular Observation Selection". In: *To appear in the Journal of Machine Learning Research* (2009) (cit. on p. 28).

[KSG08] A. Krause, A. Singh, and C. Guestrin. "Near-optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies". In: *Journal of Machine Learning Research*. Vol. 9. 2008 (cit. on p. 4).

[KW12] Lian Pin Koh and Serge A. Wich. "Dawn of drone ecology: low-cost autonomous aerial vehicles for conservation". In: *Trop Conserv Sci* (2012) (cit. on p. 127).

[Kää06] Matti Kääriäinen. "Active learning in the non-realizable case". In: *Algorithmic Learning Theory*. Springer. 2006, pp. 63–77 (cit. on pp. 36, 39).

[Lab02] Robust Image Understanding Lab. *Edge Detection and Image Segmentation (EDISON) System*. http://coewww.rutgers.edu/riul/research/code/EDISON/. 2002 (cit. on p. 129).

[LB11] Hui Lin and Jeff Bilmes. "A Class of Submodular Functions for Document Summarization". In: *NAACL/HLT-2011*. 2011 (cit. on p. 28).

[Les+07] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. "Cost-effective Outbreak Detection in Networks". In: *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2007 (cit. on p. 28).

[Les+98] Victor Lesser, Bryan Horling, Frank Klassner, Anita Raja, Thomas Wagner, and Shelley XQ. Zhang. "BIG: A Resource-Bounded Information Gathering Agent". In: *AAAI*. 1998 (cit. on p. 4).

[LGM98] Michael L. Littman, Judy Goldsmith, and Martin Mundhenk. "The Computational Complexity of Probabilistic Planning". In: *Journal of Artificial Intelligence Research* 9 (1998), pp. 1–36 (cit. on p. 4).

[Lic13] M. Lichman. *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml. 2013 (cit. on p. 157).

[Lin56] Dennis V Lindley. "On a measure of the information provided by an experiment". In: *The Annals of Mathematical Statistics* (1956), pp. 986–1005 (cit. on pp. 6, 40).

[Liu04] Ying Liu. "Active learning with support vector machine applied to gene expression data for cancer classification". In: *Journal of chemical information and computer sciences* 44.6 (2004), pp. 1936–1941 (cit. on p. 35).

[LK14] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data. June 2014 (cit. on p. 154).

[Lom+07] Rachel Lomasky, Carla E Brodley, M Aernecke, David Walt, and M Friedl. "Active class selection". In: *European Conference on Machine Learning*. Springer. 2007, pp. 640–647 (cit. on p. 38).

[Lov83] L. Lovasz. "Submodular functions and convexity". In: *Mathematical Programming - State of the Art* (1983), pp. 235–257 (cit. on p. 28).

[Lov98] Laszlo Lovasz. "Hit-and-Run Mixes Fast". In: *Math. Prog* 86 (1998), pp. 443–461 (cit. on p. 153).

[Loy+12] Chen Change Loy, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. "Stream-based joint exploration-exploitation active learning". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. 2012, pp. 1560–1567 (cit. on p. 35).

[LPW] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. *Markov chains and mixing times*. American Mathematical Soc. (cit. on p. 48).

[LS88] Steffen L. Lauritzen and David J. Spiegelhalter. "Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 50.2 (1988), pp. 157–224 (cit. on p. 22).

[Lut85] SP Luttrell. "The use of transinformation in the design of data sampling schemes for inverse problems". In: *Inverse Problems* 1.3 (1985), p. 199 (cit. on p. 8).

[Mac92a] David MacKay. "Information-based objective functions for active data selection". In: *Neural computation* 4.4 (1992), pp. 590–604 (cit. on pp. 8, 36).

[Mac92b] David J.C. MacKay. "Information-Based Objective Functions for Active Data Selection". In: *Neural Computation* 4.4 (1992), pp. 590–604 (cit. on p. 4).

[Met+53] N Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. "Equations of State Calculations by Fast Computing Machines". In: *Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092 (cit. on p. 22).

[Min78] M. Minoux. "Accelerated greedy algorithms for maximizing submodular set functions". In: *Optimization Techniques, LNCS* (1978), pp. 234–243 (cit. on p. 28).

[Mit97] Thomas M. Mitchell. *Machine Learning*. 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997 (cit. on p. 35).

[NJB05] Mukund Narasimhan, Nebojsa Jojic, and Jeff Bilmes. "Q-clustering". In: *NIPS*. 2005 (cit. on p. 28).

[NJC12] Mohammad Naghshvar, Tara Javidi, and Kamalika Chaudhuri. "Noisy bayesian active learning". In: *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE. 2012, pp. 1626–1633 (cit. on p. 41).

[NOS12] Sahand Negahban, Sewoong Oh, and Devavrat Shah. "Iterative ranking from pair-wise comparisons". In: *Advances in Neural Information Processing Systems*. 2012, pp. 2474–2482 (cit. on p. 106).

[Now08] Robert Nowak. "Generalized binary search". In: *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE. 2008, pp. 568–574 (cit. on pp. 36, 40).

[Now09] Robert Nowak. "Noisy generalized binary search". In: *Advances in neural information processing systems*. 2009, pp. 1366–1374 (cit. on p. 40).

[NW81] G. L. Nemhauser and L. A. Wolsey. "Studies on Graphs and Discrete Programming". In: North-Holland, 1981. Chap. Maximizing Submodular Set Functions: Formulations and Analysis of Algorithms, pp. 279–301 (cit. on p. 28).

[NWF78] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. "An analysis of approximations for maximizing submodular set functions—I". In: *Mathematical Programming* 14.1 (1978), pp. 265–294 (cit. on pp. 4, 8, 27, 28, 31, 137, 147, 243, 247).

[OPL97] T. Oates, M.V. Nagendra Prasad, and V. Lesser. "Cooperative Information Gathering: A Distributed Problem-Solving Approach". In: *IEE Proceedings on Software Engineering, Special Issue on Agent-based Systems* 144.1 (1997), pp. 72–88. URL: http://mas.cs.umass.edu/paper/107 (cit. on p. 4).

[ORVR13] Ian Osband, Dan Russo, and Benjamin Van Roy. "(More) efficient reinforcement learning via posterior sampling". In: *Proc. Neural Information Processing Systems (NIPS)*. 2013 (cit. on pp. 165, 167, 249).

[PGT06a] Joelle Pineau, Geoff Gordon, and Sebastian Thrun. "Anytime Point-Based Approximations for Large POMDPs". In: *JAIR* 27 (2006), pp. 335–380 (cit. on p. 4).

[PGT06b] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. "Anytime Point-based Approximations for Large POMDPs". In: *JAIR* 27.1 (Nov. 2006), pp. 335–380 (cit. on p. 34).

[PT87] C. H. Papadimitriou and J. N. Tsitsiklis. "The complexity of Markov decision processses". In: *Mathematics of Operations Research* 12.3 (1987), pp. 441–450 (cit. on p. 34).

[Puk93] F. Pukelsheim. *Optimal Design of Experiments*. John Wiley & Sons, 1993 (cit. on p. 21).

[Ray+12] Debajyoti Ray, Daniel Golovin, Andreas Krause, and Colin Camerer. "Bayesian Rapid Optimal Adaptive Design (BROAD): Method and application distinguishing models of risky choice". In: *Tech. Report* (2012) (cit. on pp. 11, 12, 105).

[RCL11] M. C. Runge, S. J. Converse, and J. E. Lyons. "Which uncertainty? Using expert elicitation and expected value of information to design an adaptive program". In: *Biological Conservation* (2011) (cit. on p. 11).

[RKJ08] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. "Learning Diverse Rankings with Multi-Armed Bandits". In: *ICML*. 2008, pp. 784–791 (cit. on pp. 4, 28).

[Rob52] H. Robbins. "Some Aspects of the Sequential Design of Experiments". In: *Bulletin of the American Mathematical Society* 58 (1952), pp. 527–535 (cit. on p. 4).

[Ros+08] Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-Draa. "Online planning algorithms for POMDPs". In: *Journal of Artificial Intelligence Research* 32 (2008), pp. 663–704 (cit. on p. 34).

[Sco10] Steven L. Scott. "A Modern Bayesian Look at the Multi-armed Bandit". In: *Appl. Stoch. Model. Bus. Ind.* 26.6 (2010), pp. 639–658 (cit. on p. 165).

[SDW01] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. "Active Learning of Partially Hidden Markov Models for Information Extraction". In: *ECML/PKDD Workshop on Instance Selection*. 2001 (cit. on p. 4).

[Set12] Burr Settles. "Active learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.1 (2012), pp. 1–114 (cit. on pp. 6, 35).

[SGB05] C. Stachniss, G. Grisetti, and W. Burgard. "Information Gain-based Exploration Using Rao-Blackwellized Particle Filters". In: *RSS*. 2005 (cit. on p. 4).

[Sha+15] Nihar B Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, and Martin J Wainwright. "Estimation from Pairwise Comparisons: Sharp Minimax Bounds with Topology Dependence." In: *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2015 (cit. on p. 106).

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423 (cit. on pp. 9, 25).

[Sha64] William F. Sharpe. "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk". In: *The Journal of Finance* (1964) (cit. on p. 104).

[Sin01] Amit Singhal. "Modern Information Retrieval: A Brief Overview". In: *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 24.4 (2001), pp. 35–43 (cit. on p. 4).

[Smi84] Robert L. Smith. "Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions". English. In: *Operations Research* (1984) (cit. on p. 153).

[SN11] M. Seeger and H. Nickisch. "Large Scale Bayesian Inference and Experimental Design for Sparse Linear Models". In: *SIAM Journal on Imaging Sciences* 4.1 (2011), pp. 166–199 (cit. on pp. 4, 21).

[Som+13] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. "Despot: Online pomdp planning with regularization". In: *Advances In Neural Information Processing Systems*. 2013, pp. 1772–1780 (cit. on p. 34).

[SR05] R. Sim and N. Roy. "Global A-Optimal Robot Exploration in SLAM". In: *ICRA*. 2005 (cit. on p. 4).

[SS73] R.D. Smallwood and E.J. Sondik. "The optimal control of partially observable Markov decision processes over a finite horizon". In: *Operations Research* 21 (1973), pp. 1071–1088 (cit. on pp. 4, 34).

[SV10] David Silver and Joel Veness. "Monte-Carlo planning in large POMDPs". In: *Advances in neural information processing systems*. 2010, pp. 2164–2172 (cit. on p. 34).

[SW97] P. Sebastiani and H. Wynn. "Bayesian experimental design and Shannon information". In: *Proceedings of the Section on Bayesian Statistical Science*. 1997, pp. 176–181 (cit. on p. 21).

[SZ96] Katia Sycara and Dajun Zeng. "Multi-Agent Integration of Information Gathering and Decision Support". In: *European Conference on Artificial Intelligence*. 1996 (cit. on p. 4).

[TC01] Simon Tong and Edward Chang. "Support vector machine active learning for image retrieval". In: *Proceedings of the ninth ACM international conference on Multimedia*. ACM. 2001, pp. 107–118 (cit. on p. 35).

[Tho33] William R. Thompson. "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples". In: *Biometrika* 25.3/4 (1933), pp. 285–294 (cit. on p. 165).

[THTS05] Gokhan Tur, Dilek Hakkani-Tür, and Robert E Schapire. "Combining active and semi-supervised learning for spoken language understanding". In: *Speech Communication* 45.2 (2005), pp. 171–186 (cit. on p. 35).

[TK01a] Simon Tong and Daphne Koller. "Active Learning for Parameter Estimation in Bayesian Networks". In: *NIPS*. 2001 (cit. on pp. 4, 26, 35).

[TK01b] Simon Tong and Daphne Koller. "Support vector machine active learning with applications to text classification". In: *Journal of machine learning research* 2.Nov (2001), pp. 45–66 (cit. on p. 35).

[TK92] Amos Tversky and Daniel Kahneman. "Advances in prospect theory: Cumulative representation of uncertainty". In: *Journal of Risk and Uncertainty* 5.4 (1992) (cit. on p. 104).

[Tsy04] Alexandre B Tsybakov. "Optimal aggregation of classifiers in statistical learning". In: *Annals of Statistics* (2004), pp. 135–166 (cit. on pp. 37, 39).

[Van+15] Hastagiri P Vanchinathan, Andreas Marfurt, Charles-Antoine Robelin, Donald Kossmann, and Andreas Krause. "Discovering valuable items from massive data". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015, pp. 1195–1204 (cit. on p. 37).

[VG11] Sudheendra Vijayanarasimhan and Kristen Grauman. "Large-scale live active learning: Training object detectors with crawled data and crowds". In: *CVPR*. 2011 (cit. on p. 137).

[VKR08] S. Vicente, V. Kolmogorov, and C. Rother. "Graph Cut based image segmentation with connectivity priors". In: *CVPR*. 2008 (cit. on p. 28).

[Wak10] P.P. Wakker. *Prospect Theory: For Risk and Ambiguity*. Cambridge University Press, 2010 (cit. on p. 104).

[WFIW07] Jason L Williams, John W. Fisher III, and Alan S Willsky. "Performance guarantees for information theoretic active inference". In: *International Conference on Artificial Intelligence and Statistics*. 2007, pp. 620–627 (cit. on p. 41).

[WG75] Peter M. Will and David D. Grossman. "An Experimental System for Computer Controlled Mechanical Assembly". In: *IEEE Trans. Computers* 24.9 (1975), pp. 879–888 (cit. on pp. 7, 111).

[WGS13] Xuezhi Wang, Roman Garnett, and Jeff G. Schneider. "Active search on graphs." In: *KDD*. 2013, 731–738 (cit. on p. 37).

[Wol82] Laurence A. Wolsey. "An analysis of the greedy algorithm for the submodular set covering problem". In: *Combinatorica* 2.4 (1982), pp. 385–393 (cit. on p. 32).

[WP67] D. J. WELSH and M. B. POWELL. "An Upper Bound for the Chromatic Number of a Graph and Its Application to Timetabling Problems". In: *Computer Journal* (1967) (cit. on p. 86).

[YC13] Liu Yang and Jaime Carbonell. "Buy-in-bulk active learning". In: *Advances in Neural Information Processing Systems*. 2013, pp. 2229–2237 (cit. on pp. 37, 152).

[YJ08] Yisong Yue and T. Joachims. "Predicting Diverse Subsets Using Structural SVMs". In: *International Conference on Machine Learning (ICML)*. 2008, pp. 271–278 (cit. on p. 28).

[ZC14] Chicheng Zhang and Kamalika Chaudhuri. "Beyond disagreement-based agnostic active learning". In: *Advances in Neural Information Processing Systems*. 2014, pp. 442–450 (cit. on pp. 37, 39, 40).

[Zhu15] Xiaojin Zhu. "Machine Teaching: An Inverse Problem to Machine Learning and an Approach Toward Optimal Education." In: *AAAI*. 2015, pp. 4083–4087 (cit. on p. 179).

[ZL96] Shlomo Zilberstein and Victor Lesser. *Intelligent Information Gathering Using Decision Models*. Tech. rep. Computer Science Department, University of Massachusetts, 1996 (cit. on p. 4).

[ZRB05] Alice X. Zheng, Irina Rish, and Alina Beygelzimer. "Efficient Test Selection in Active Diagnosis via Entropy Approximation". In: *In Proceedings of UAI-05*. 2005 (cit. on p. 9).

[ZSR02] F. Zhao, J. Shin, and J. Reich. "Information-Driven Dynamic Sensor Collaboration for Tracking Applications". In: *IEEE Signal Processing* 19.2 (2002), pp. 61–72 (cit. on p. 4).

[Ünl04] Tonguç Ünlüyurt. "Sequential testing of complex systems: a review". In: *Discrete Applied Mathematics* 142.1 (2004), pp. 189–205 (cit. on p. 41).