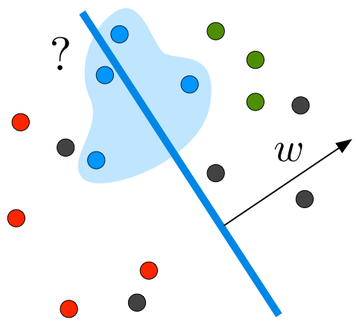
Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization

Yuxin Chen and Andreas Krause, ETH Zurich, Switzerland

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

TASK: BATCH SELECTION



Pool-based Batch Mode Active Learning

amazon mechanical turk beta

Multi-stage Influence Maximization in Social Networks

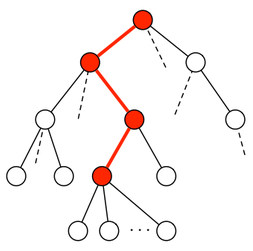
FORMULATION

Items: $\mathcal{V} = \{1, \ldots, n\}$ RV: $\mathbf{Y}_\mathcal{V} = [Y_1, \ldots, Y_n]$

Item-realization Pairs: $\mathbf{y}_\mathcal{A}$, $\mathcal{S}(\pi, \mathbf{y}_\mathcal{V}) \subseteq \mathcal{V} \times O$

Policy

$$\pi : 2^{\mathcal{V} \times O} \to \mathcal{V}$$



Objective function

$$f : 2^{\mathcal{V} \times O} \to \mathbb{N}.$$

Cost of a policy

$$\text{cost}_{ac}(\pi) = \mathbb{E}_{\mathbf{y}_\mathcal{V}}[|\mathcal{S}(\pi, \mathbf{y}_\mathcal{V})|]$$
$$\text{cost}_{wc}(\pi) = \max_{\mathbf{y}_\mathcal{V}} |\mathcal{S}(\pi, \mathbf{y}_\mathcal{V})|$$

SEQUENTIAL: MIN-COST COVER

$$\min_{\pi \in \Pi} \text{cost}(\pi), \quad \text{s.t.}$$
$$f(\mathcal{S}(\pi, \mathbf{y}_\mathcal{V})) \geq Q \text{ for all } \mathbf{y}_\mathcal{V} \text{ with } P(\mathbf{y}_\mathcal{V}) > 0.$$

THE OBJECTIVE FUNCTION

✓ Normalized:
- We derive no utility from knowing nothing $f(\emptyset) = 0$

✓ Monotonic:
- Adding labels never hurts $\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{V} \times O \Rightarrow f(\mathcal{S}) \leq f(\mathcal{S}')$

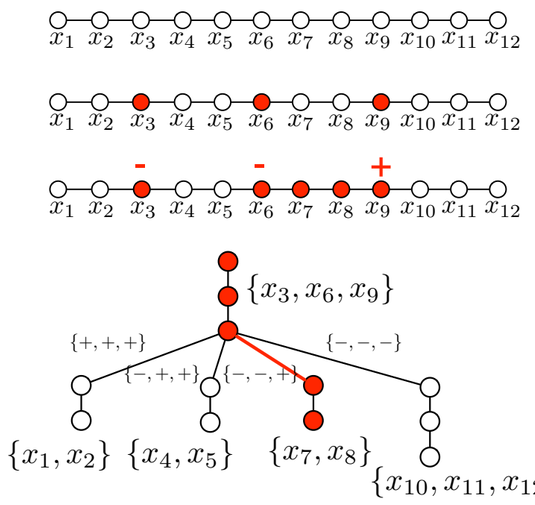
✓ Submodular
- Adding a label helps more if we have observed less labels $\left\{ \begin{array}{l} \mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{V} \times O \\ (j, y) \in \mathcal{V} \times O \setminus \mathcal{S}' \end{array} \right. \Rightarrow f(\mathcal{S} \cup \{(j,y)\}) - f(\mathcal{S}) \geq f(\mathcal{S}' \cup \{(j,y)\}) - f(\mathcal{S}')$

✓ Adaptive Submodular
- The gain of an item, in expectation over its unknown label, can never increase as we gather more information: $\left\{ \begin{array}{l} \mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{V} \times O \\ P(\mathcal{S}') > 0 \end{array} \right. \Rightarrow \Delta_f(j \mid \mathcal{S}) \geq \Delta_f(j \mid \mathcal{S}')$

Conditional Marginal Gain

$$\Delta_f(j \mid \mathcal{S}) = \sum_y P(Y_j = y \mid \mathcal{S}) \Big[f(\mathcal{S} \cup \{(j,y)\}) - f(\mathcal{S}) \Big].$$

EX: BMAL -- 1-D THRESHOLD



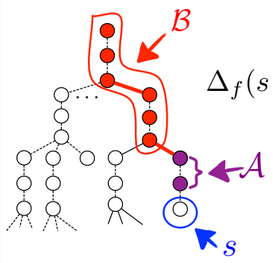
Unlabeled data

1st selected batch

2nd selected batch

The decision tree representing the BatchGreedy policy

THE BATCHGREEDY ALGORITHM



Conditional marginal benefit of an item s:

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

Expectation over all realizations within the batch

The utility of a specific realization

Conditioning on previous observations

The BatchGreedy policy will greedily select the i-th element in the j-th batch

$$s_{i,j} = \arg\max_{s \in \mathcal{V}} \Delta_f(s \mid \underbrace{\{s_{1,j}, \ldots, s_{i-1,j}\}}_{\text{the } j_{th} \text{ batch } \mathcal{A}}, \mathbf{y}_\mathcal{B})$$

BATCHGREEDY VS. OPTIMAL BATCH

$$\mathbb{E}\left|\text{BatchGreedy}\right| \leq \mathbb{E}\left|\text{Optimal}\right| \cdot \left(\frac{e}{e-1}\right)\left(\ln Q + 1\right)$$

Expected cost of **BatchGreedy**

Expected cost of **optimal** policy selecting batches of size k

$$\max\left|\text{BatchGreedy}\right| \leq \max\left|\text{Optimal}\right| \cdot \left(\frac{e}{e-1}\right)\left(\ln \frac{Q}{\delta} + 1\right)$$

Worst-case cost of **BatchGreedy**

Worst-case cost of **optimal** policy selecting batches of size k

Minimum probability of any realization (i.e., branch)

PRIOR WORK: NON-ADAPTIVE VS. SEQ.

MIN-COST Cover

The expected **cost** incurred by seq. policy to achieve Q can be **arbitrarily lower** than the cost of the best non-ada. policy.

[Goemans and Vondrak '06]

MAX Cover

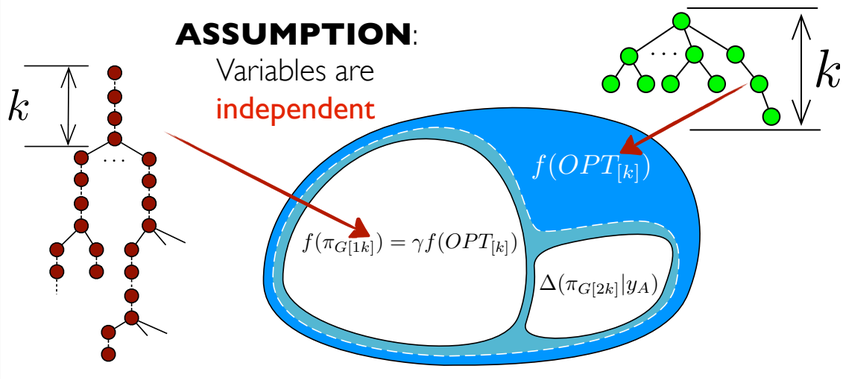
The **utility** of seq. algorithm (with cardinality cstr.) **cannot be much higher** than that of the best non-ada. algorithm.

[Asadpour et al, '08]

OUR RESULT: BATCH-ADAPTIVE VS. SEQ.

Under some constraints, **BatchGreedy** is competitive with the optimal **sequential** policy.

ASSUMPTION: Variables are independent



BATCHGREEDY VS. OPTIMAL SEQ.

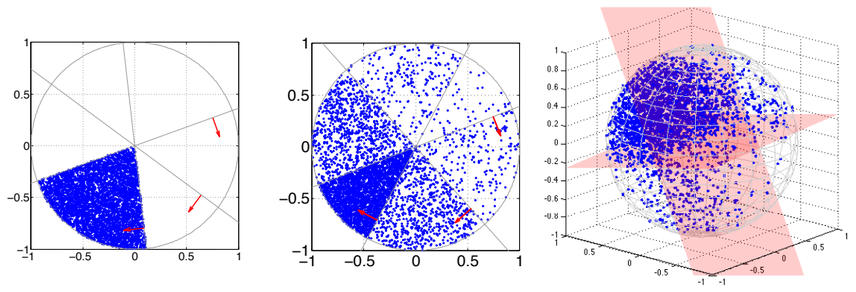
Theorem Fix $\beta > 0$. Let OPT_{wc} be the worst-case cost of an optimal **sequential** policy constrained to picking a number of items which is a multiple of k. Further suppose that the variables $Y_1, \ldots, Y_n$ are *independent*. Then for the cost of the BATCHGREEDY policy π_G, run until it achieves $f(\pi_G) \geq Q - \beta$,

$$\text{cost}_{wc}(\pi_G) \leq \text{OPT}_{wc} \left(e/(e-1)\right)^2 \left(\ln \frac{Q}{\beta} + 1\right).$$

Moreover, $P\big(f(\mathcal{S}(\pi_G, \mathbf{y}_\mathcal{V})) \geq Q\big) \geq 1 - \beta$.

VERSION SPACE SAMPLING

ASSUMPTION: Hypotheses that violate more cstr. induce lower confidence.



MCMC sampler

Hit-and-run sampler for linear separators and noisy observations

BMAL: APPROXIMATE IMPLEMENTATION

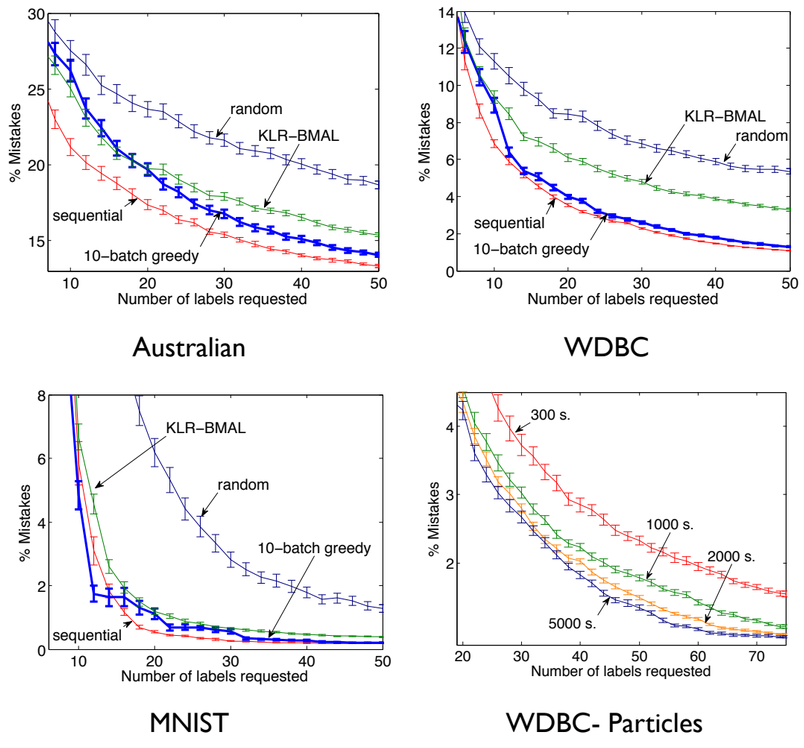
Step 1: Sample the version space

Step 2: For each iteration within the batch, pick

$$s \leftarrow \arg\min_{s'} \sum_{\ell=1}^{N} \Big[\hat{P}(\underbrace{\mathcal{H}(\{(\mathbf{x}, h_\ell(\mathbf{x})) : \mathbf{x} \in \mathcal{A} \cup \{s'\}\})}_{\text{hypotheses that agree with } h_l \text{ .}}) \Big]$$

Step 3: At the end of each batch, observe the labels, go to Step 1.

EXPERIMENTS: ACTIVE LEARNING



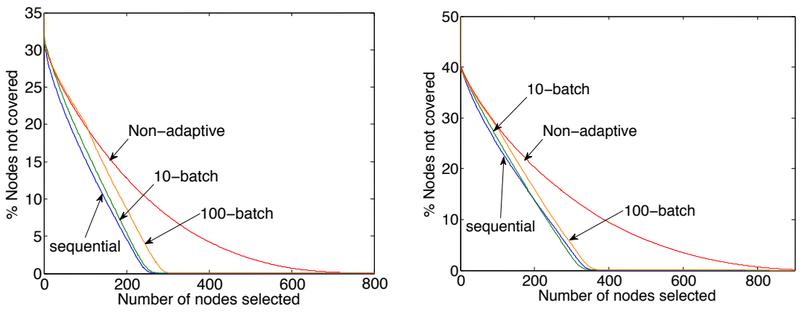
Australian

WDBC

MNIST

WDBC- Particles

EXPERIMENTS: MAX INFLUENCE



Epinions

Slashdot

Code

www.inf.ethz.ch/~chenyux/icml13/bmal-src.zip