Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization

Yuxin Chen and Andreas Krause, ETH Zurich, Switzerland

**TASK: BATCH SELECTION**

Pool-based Batch Mode Active Learning

- Adaptive Submodular Optimization

**FOMULATION**

Items: \( V = \{1, \ldots, n\} \)  
RV: \( Y = [Y_1, \ldots, Y_n] \)

Item-realization Pairs: \( Y_A, \mathcal{S}(\pi, y_V) \subseteq V \times O \)

Policy \( \pi : 2^V \rightarrow V \)

Objective function \( f : 2^V \rightarrow \mathbb{R} \)

Cost of a policy \( c_{\omega}(\pi) = \mathbb{E}_y[|\mathcal{S}(\pi, y_V)|] \)

**SEQUENTIAL: MIN-COST COVER**

\[
\min_{\pi \in \Pi} c_{\omega}(\pi), \ s.t. \ f(\mathcal{S}(\pi, y_V)) \geq Q \text{ for all } y_V \text{ with } P(y_V) > 0.
\]

**THE OBJECTIVE FUNCTION**

- Normalized: \( f(\emptyset) = 0 \)
- Monotonic: Adding labels never hurts \( f(S) \leq f(S') \)
- Submodular: Adding a label helps more if we have observed less labels
- Adaptive Submodular: The gain of an item, in expectation over its unknown label, can never increase as we gather more information:

  \[
  \Delta_f(j|S) = \sum_y P(Y_j = y | S) f(S \cup \{j, y\}) - f(S)
  \]

**EX: BMAL -- 1-D THRESHOLD**

- 1st selected batch \( \{x_1, x_2\} \)
- 2nd selected batch \( \{x_3, x_6, x_9\} \)

**THE BATCHGREEDY ALGORITHM**

The conditional marginal benefit of an item \( j \) is defined as:

\[
\Delta_f(j|S) = \mathbb{E}_y[f(S \cup \{j, y\}) - f(S)].
\]

The BatchGreedy policy will greedily select the \( j \)-th element in the \( j \)-th batch

\[
s_{i,j} = \arg\max_{y} \Delta_f(j | \mathcal{S}(\{s_{i,j}, \ldots, s_{i-1,j}\}, y))
\]

**THEOREM**

If \( \Delta_f(j|S) \geq \Delta_f(i|S) \) for all \( i < j \), then

\[
\mathbb{E}[f(\mathcal{S}(\pi, y_V)) | \mathcal{S}(\pi, y_V) \neq \emptyset] = \mathbb{E}[f(\mathcal{S}(\pi', y_V)) | \mathcal{S}(\pi', y_V) \neq \emptyset]
\]

**EXPERIMENTS: MAX INFLUENCE**

- MNIST
- WDBC
- Epinions
- Code
- Slashdot

**ASSUMPTION**

Variables are independent

\[
f(\mathcal{O}_k) = \sum_{k=0}^{\infty} P(Y_k = y_k | S) f(S \cup \{k, y_k\}) - f(S)
\]

**OUR RESULT: BATCH-ADAPTIVE VS. SEQ.**

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