Near-optimal Adaptive Pool-based Active Learning with General Loss

Nguyen Viet Cuong   Wee Sun Lee   Nan Ye

Department of Computer Science, National University of Singapore

Table of Contents

Introduction (1)

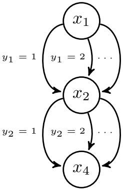
- Pool-based active learning:
  - Select training data sequentially from a finite pool of unlabeled examples
  - *Adaptive selection*: selected example is labeled before next example is chosen
  - *Aim*: good performance with small number of labeled examples
  - *Fixed-budget version*: select k examples that optimize some objective (utility).
- Common method: greedy algorithms
  - Query the example optimizing some criterion (1-step utility).
  - *Analysis:* near-optimality of coverage
    - Utility of k selected examples is within a constant factor of the optimal utility.

Introduction (2)

- Examine three greedy algorithms:
  - Maximum entropy [Settles, 2010]
    - No constant factor approximation in average case
  - Least confidence [Lewis and Gale, 1994]
    - Constant factor approximation in worst case
  - Maximum Gibbs error [Cuong et al., 2013]
    - Constant factor approximation in average case
- Generalize maximum Gibbs error with loss functions
  - *Average case*: constant factor approximation if utility is adaptive submodular
  - *Worst case*: constant factor approximation with any loss
- Worst-case results are special cases of new general result for *pointwise submodular functions*.

Notations

- $\mathcal{X}$: pool of unlabeled examples, $x \in \mathcal{X}$
- $\mathcal{Y}$: label set, $y \in \mathcal{Y}$
- A (partial) labeling: a (partial) function from $\mathcal{X}$ to $\mathcal{Y}$.
- Hypothesis space $\mathcal{H}$ consists of all labelings.
- Bayesian setting: prior $p_0[h]$ on $\mathcal{H}$, posterior $p_{\mathcal{D}}$
- An active learning algorithm is a *policy* for choosing examples $\longrightarrow$ represented by a *policy tree*



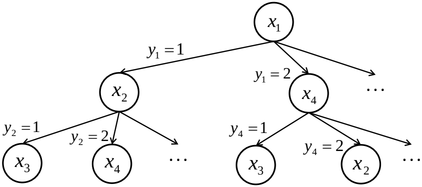
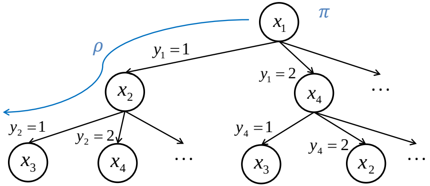


Table of Contents

Maximum Entropy (1)

- What we want: after choosing k examples, minimize entropy of the remaining examples
  - By maximizing entropy of the selected examples
  - Greedy is near-optimal in *non-adaptive* case: $(1 - \frac{1}{e})$-factor approximation
- Objective in adaptive case: maximize the *policy entropy*

$$H_{\mathrm{ent}}(\pi) \stackrel{\mathrm{def}}{=} \mathbb{E}_{\rho}[-\ln p_0[\rho]]$$



- ρ: a path in the policy tree
- $p_0[\rho]$: probability that π follows path ρ
- Each path contains selected examples and their labels.
- Entropy over paths

Maximum Entropy (2)

- Usual *maximum entropy* algorithm is a natural greedy method to maximize policy entropy
- Greedy criterion: $x_{\text{next}} = \arg\max_x \mathbb{E}_{y \sim p_{\mathcal{D}}[y;x]}[-\ln p_{\mathcal{D}}[y;x]]$
  - $p_{\mathcal{D}}[y;x]$: posterior probability that y is label of x
- Is there a performance guarantee for greedy?

Theorem (Non-optimality)

For $0 < \alpha < 1$, there exists a problem where $\dfrac{H_{ent}(\pi_{greedy})}{H_{ent}(\pi_{optimal})} < \alpha$.

Least Confidence (1)

- Greedy criterion: $x_{\text{next}} = \arg\min_x \{\max_y p_{\mathcal{D}}[y; x]\}$
- Objective: maximize worst-case version space reduction

$$H_{\text{lc}}(\pi) \stackrel{\text{def}}{=} \min_h f(x_h, h)$$

  - x_h: examples selected by π under true labeling h
  - f: version space reduction

$$f(S, h) \stackrel{\text{def}}{=} 1 - p_0[h(S); S]$$

  - $p_0[h(S); S] \equiv$ probability that $h(S)$ is the label sequence of S $\equiv$ volume of the remained labelings in version space, weighted by probability

Least Confidence (2)

Theorem (Near-optimality)

$$H_{lc}(\pi_{greedy}) > \left(1 - \frac{1}{e}\right) H_{lc}(\pi_{optimal}) \approx 0.6\, H_{lc}(\pi_{optimal})$$

- Special case of the next general result for pointwise submodular functions.

Pointwise Submodular Functions

- Submodular function: $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$ for $A \subseteq B \subseteq \mathcal{X}$ and $x \in \mathcal{X} \setminus B$.
- Monotone function: $f(A) \leq f(B)$ for $A \subseteq B$.

Definition (Pointwise submodular function)

Utility function $f(S, h)$ is pointwise submodular if the set function $f_h(S) \stackrel{\text{def}}{=} f(S, h)$ is submodular for all h.

Definition (Pointwise monotone function)

f is pointwise monotone if $f_h(S)$ is monotone for all h.

Definition (Minimal dependency)

$f(S, h)$ does *not* depend on labels of $\mathcal{X} \setminus S$.

Maximizing Pointwise Submodular Functions

- Greedy criterion: maximize *worst-case utility gain*

$$x_{\text{next}} = \arg\max_x \delta(x|\mathcal{D}), \text{ where}$$

$$\delta(x|\mathcal{D}) \stackrel{\text{def}}{=} \min_y \{f(\text{domain}(\mathcal{D}) \cup \{x\}, \mathcal{D} \cup \{(x, y)\}) - f(\text{domain}(\mathcal{D}), \mathcal{D})\}$$

- Worst-case objective: maximize $f_{\text{worst}}(\pi) \stackrel{\text{def}}{=} \min_h f(x_h, h)$

Theorem (Near-optimality)

If f is pointwise monotone submodular and satisfies minimal dependency, then $f_{worst}(\pi_{greedy}) > \left(1 - \frac{1}{e}\right) f_{worst}(\pi_{optimal})$.

- Version space reduction satisfies these conditions.

Maximum Gibbs Error (1)

- Recall policy entropy: $H_{\text{ent}}(\pi) = \mathbb{E}_\rho[-\ln p_0[\rho]]$
- Maximize a lower bound of $H_{\text{ent}}(\pi)$ instead
- Objective: policy Gibbs error

$$H_{\text{gibbs}}(\pi) \stackrel{\text{def}}{=} \mathbb{E}_\rho[1 - p_0[\rho]]$$

- Greedy criterion: maximum Gibbs error

$$x_{\text{next}} = \arg\max_x \mathbb{E}_{y \sim p_{\mathcal{D}}[y;x]}[1 - p_{\mathcal{D}}[y;x]]$$

- $1 - p_{\mathcal{D}}[y;x] \equiv$ probability that a random label drawn from $p_{\mathcal{D}}[\,\cdot\,;x]$ is not $y \equiv$ error rate of a Gibbs classifier
- $H_{\text{gibbs}}(\pi)$ is expected version space reduction after running π.
- This criterion is equivalent to greedily maximize expected version space reduction.

Maximum Gibbs Error (2)

- Maximum Gibbs error guarantees constant-factor approximation for maximizing $H_{\text{gibbs}}(\pi)$ [Cuong et al., 2013].

$$H_{\text{gibbs}}(\pi_{\text{greedy}}) > \left(1 - \frac{1}{e}\right) H_{\text{gibbs}}(\pi_{\text{optimal}})$$

- Due to the *adaptive submodularity* of version space reduction [Golovin and Krause, 2011].

Table of Contents

Generalized Version Space Reduction

- Recall version space reduction

$$f(S,h) = 1 - p_0[h(S); S] = \mathbb{E}_{h' \sim p_0}[\mathbf{1}(h(S) \neq h'(S))]$$

- Disadvantage: Expected 0-1 loss that a random labeling of S differs from $h(S)$
- Improvement: *generalized version space reduction*

$$f_L(S,h) \stackrel{\text{def}}{=} \mathbb{E}_{h' \sim p_0}[L(h,h')\ \mathbf{1}(h(S) \neq h'(S))]$$

  - L is a loss function. For example,
    - Hamming loss: $L(h,h') = \sum_{x \in \mathcal{X}} |h(x) - h'(x)|$
    - 0-1 loss: $L(h,h') = \mathbf{1}(h \neq h')$
  - For 0-1 loss, $f_L(S,h)$ = normal version space reduction

The Average-case Criterion

- Objective: maximize expected value of $f_L(S, h)$

$$H_L^{\mathrm{avg}}(\pi) \stackrel{\mathrm{def}}{=} \mathbb{E}_{h \sim p_0}[f_L(x_h, h)]$$

- Greedy criterion: maximize expected utility gain

$$x_{\mathrm{next}} = \arg\max_x \mathbb{E}_{h \sim p_{\mathcal{D}}}[f_L(\mathrm{domain}(\mathcal{D}) \cup \{x\}, h) - f_L(\mathrm{domain}(\mathcal{D}), h)]$$

- $(1 - \frac{1}{e})$-factor guarantee if f_L is adaptive monotone submodular.
- However, f_L may not always be adaptive submodular.
- Near-optimality guarantee in general is still unknown.

The Worst-case Criterion

- Total generalized version space reduction

$$t_L(S,h) \stackrel{\text{def}}{=} \sum_{h',h''} p_0[h']L(h',h'')p_0[h''] - \sum_{h',h'':h'(S)=h''(S)=h(S)} p_0[h']L(h',h'')p_0[h'']$$

- Pointwise monotone submodular and minimal dependency
- Objective: maximize $T_L^{\text{worst}}(\pi) \stackrel{\text{def}}{=} \min_h t_L(x_h, h)$
- Greedy criterion: maximize worst-case utility gain

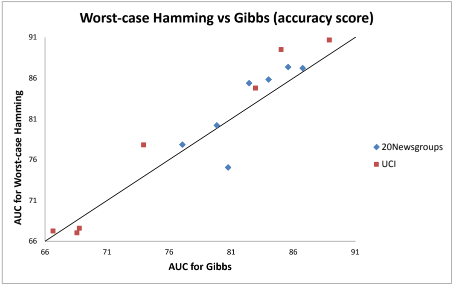
$$x_{\text{next}} = \arg\max_x \{\min_y [t_L(\text{domain}(\mathcal{D}) \cup \{x\}, \mathcal{D} \cup \{(x,y)\}) - t_L(\text{domain}(\mathcal{D}), \mathcal{D})]\}$$

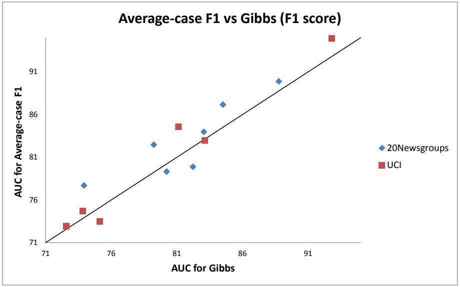
Theorem (Near-optimality)

$$T_L^{worst}(\pi_{greedy}) > \left(1 - \frac{1}{e}\right) T_L^{worst}(\pi_{optimal}).$$

Table of Contents

Experimental Results





- Binary-class problem: maximum Gibbs error, maximum entropy, and least confidence are equivalent
- Also equivalent to using 0-1 loss

Conclusion

- New near-optimality guarantee for maximizing pointwise submodular functions in worst case
- New theoretical properties of maximum entropy and least confidence
- Two new active learning algorithms with general loss
  - Easy to approximate
  - One algorithm is near-optimal in worst case.

Thank you.