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

Nguyen Viet Cuong
Department of Computer Science
National University of Singapore
nvcuong@comp.nus.edu.sg

Wee Sun Lee
Department of Computer Science
National University of Singapore
leews@comp.nus.edu.sg

Nan Ye
Department of Computer Science
National University of Singapore
yenan@comp.nus.edu.sg

Abstract

We consider adaptive pool-based active learning in a Bayesian setting. We first analyze two commonly used greedy active learning criteria: the maximum entropy criterion, which selects the example with the highest entropy, and the least confidence criterion, which selects the example whose most probable label has the least probability value. We show that unlike the non-adaptive case, the maximum entropy criterion is not able to achieve an approximation that is within a constant factor of optimal policy entropy. For the least confidence criterion, we show that it is able to achieve a constant factor approximation to the optimal version space reduction in a worst-case setting, where the probability of labelings that have not been eliminated is considered as the version space. We consider a third greedy active learning criterion, the Gibbs error criterion, and generalize it to handle arbitrary loss functions between labelings. We analyze the properties of the generalization and its variants, and show that they perform well in practice.

1 INTRODUCTION

We study pool-based active learning (McCallum and Nigam, 1998) where the training data are sequentially selected and labeled from a pool of unlabeled examples, with the aim of having good performance after only a small number of examples are labeled. In practice, the selection of the next example to be labeled is usually done by greedy optimization of some appropriate objective function.

In this paper, we consider adaptive algorithms for pool-based active learning with a budget of $k$ queries in a Bayesian setting. We examine three commonly used greedy criteria and their performance guarantees. We also generalize one of the criteria, study its properties and show that it performs well in practice.

One of the most commonly used criteria is the maximum entropy criterion: select the example with maximum label entropy given the observed labels (Settles, 2010). In the non-adaptive case where the set of examples must be selected before any label is observed, the analogue of this greedy criterion selects the example that maximally increases the label entropy of the selected set. This greedy criterion in the non-adaptive case is well-known to be near-optimal: the label entropy of the selected examples is at least $(1 - 1/e)$ of the optimal set. This follows from a property satisfied by the entropy function called submodularity. Selecting a set with large label entropy is desirable, as the chain rule of entropy implies that maximizing the label entropy of the selected set will minimize the conditional label entropy of the remaining examples. It would be desirable to have a similar near-optimal performance guarantee for the adaptive case where the label is provided after every example is selected. Whether the greedy maximum entropy criterion provides such a guarantee was not known (Cuong et al., 2013), although it was suspected that it does not. In this paper, we show that the greedy algorithm, indeed, does not provide a constant factor approximation in the adaptive case.

Another commonly used greedy criterion is the least confidence criterion: select the example whose most likely label has the smallest probability (Lewis and Gale, 1994; Culotta and McCallum, 2005). In this paper, we show that this criterion provides a near-optimal adaptive algorithm for maximizing the worst-case version space reduction, where the version space is the probability of labelings that are consistent with the observed labels. This will be derived as the consequence of a more general result which shows such near-optimal approximation holds for utility functions that satisfy pointwise submodularity and minimal dependency. Pointwise submodular functions were previously studied in (Guillory and Bilmes, 2010) for active learning, but with a different objective function which focuses on identifying the true function.

The Gibbs error criterion was proposed in (Cuong et al., 2013) as an alternative uncertainty measure suitable for ac-
tive learning. The criterion selects the example with the largest Gibbs error for labeling. The Gibbs error is the expected error of the Gibbs classifier, which predicts the label by sampling from the current label distribution. Gibbs error is a special case of Tsallis entropy, introduced in statistical mechanics (Tsallis and Brigatti, 2004) as a generalization of the Shannon entropy (which is used in the maximum entropy criterion). In (Cuong et al., 2013), Gibbs error was used as a lower bound to the Shannon entropy and was maximized in order to minimize the posterior conditional entropy. It was shown in (Cuong et al., 2013) that using the Gibbs error criterion achieves at least \((1 - 1/e)\) of the optimal policy Gibbs error, a performance measure for this criterion, given \(k\) queries in the adaptive case. This relies on the property that the version space reduction function is adaptive submodular (Golovin and Krause, 2011).

The results for the three commonly used greedy criteria are shown in Table 1.

The Gibbs error criterion can be seen as a greedy algorithm for sequentially maximizing the Gibbs error over the dataset. The Gibbs error of the dataset is the expected error of a Gibbs classifier that predicts using an entire labeling sampled from the prior label distribution for the entire dataset. Here, a labeling is considered incorrect if any example is incorrectly labeled by the Gibbs classifier. Predicting an entirely correct labeling of all examples is often unrealistic in practice, particularly after only a few examples are labeled. This motivates us to generalize the Gibbs error to handle different loss functions between labelings, e.g. Hamming loss which measures the Hamming distance between two labelings. We call the greedy criterion that uses general loss functions the average generalized Gibbs error criterion.

The corresponding performance measure for the average generalized Gibbs error criterion is the generalized policy Gibbs error, which is the expected value of the generalized version space reduction function. The generalized version space reduction function is an extension of the version space reduction function with general loss functions. We investigate whether the generalized version space reduction function is adaptive submodular, as this property would provide a constant factor approximation for the average generalized Gibbs error criterion. Unfortunately, we can show that the generalized version space reduction function is not necessarily adaptive submodular, although it is adaptive submodular for the special case of the version space reduction function. Despite that, we show in our experiments that the average generalized Gibbs error criterion can perform well in practice, even when we do not know whether the corresponding utility function is adaptive submodular.

As in the case for the least confidence criterion, we also consider a worst-case setting for the generalized Gibbs error. The worst case against a target labeling can be severe, so we consider a variant: the total generalized version space reduction function. This function targets the sum of the remaining losses over all the remaining labelings, rather than against a single worst-case labeling. We call the corresponding greedy criterion the worst-case generalized Gibbs error criterion. It selects the example with maximum worst-case total generalized version space reduction as the next query. As the total generalized version space reduction function is pointwise submodular and satisfies the minimal dependency property, the method is guaranteed to be near-optimal. Our experiments show that the worst-case generalized Gibbs error criterion performs well in practice. For binary problems, the maximum entropy, least confidence, and Gibbs error criteria are all equivalent, and the worst-case generalized Gibbs error criterion outperforms them for most problems in our experiments.

### 2 PRELIMINARIES

Let \(\mathcal{X}\) be a finite set of items (or examples), and let \(\mathcal{Y}\) be a finite set of labels (or states). A labeling of \(\mathcal{X}\) is a function from \(\mathcal{X}\) to \(\mathcal{Y}\), and a partial labeling is a partial function from \(\mathcal{X}\) to \(\mathcal{Y}\). Each labeling of \(\mathcal{X}\) can be considered as a hypothesis in the hypothesis space \(\mathcal{H} = 2^\mathcal{X}\). In the Bayesian setting, there is a prior probability \(p_0[h]\) on \(\mathcal{H}\), and an unknown true hypothesis \(h_{true}\) is initially drawn from \(p_0[h]\). After observing a labeled set (i.e., a partial labeling) \(D\),
we can obtain the posterior \( p_D[h] = p_0[h|D] \) using Bayes’ rule.

For any \( S \subseteq \mathcal{X} \) and any distribution \( p \) on \( \mathcal{H} \), we write \( p[y; S] \) to denote the probability that a randomly drawn hypothesis from \( p \) assigns labels in the sequence \( y \) to items in the sequence \( S \). That is, \( p[y; S] \equiv \sum_{h \in \mathcal{H}} p[h] \mathbb{P}[h(S) = y|h] \), where we use the notation \( h(S) \) to denote the sequence \( (h(x_1), \ldots, h(x_i)) \) whenever \( S \) is a sequentially constructed set \((x_1, \ldots, x_i)\), or simply the set \( \{h(x) : x \in S\} \) if the items in \( S \) are not ordered. In our setting, \( h \) is a deterministic hypothesis, so \( \mathbb{P}[h(S) = y|h] = 1(h(S) = y) \), where \( 1(\cdot) \) is the indicator function. Note that \( p[y; S] \) is a probability distribution on the set of all label sequences \( y \) of \( S \). For \( x \in \mathcal{X} \) and \( y \in \mathcal{Y} \), we also write \( p[y; x] \) for \( p[y; \{x\}] \).

In practice, we often consider probabilistic models (like the naive Bayes models) which are used to generate labels for examples, and a prior is imposed on these models instead of on the labelings. In this case, we can follow the construction in the supplementary material of (Cuong et al., 2013) to construct an equivalent prior on the labelings and work with this induced prior.

We consider pool-based active learning with a fixed budget: given a budget of \( k \) queries, we aim to adaptively select from the pool \( \mathcal{X} \) the best \( k \) examples with respect to some objective function.\(^1\) A pool-based active learning algorithm corresponds to a policy for choosing training examples from \( \mathcal{X} \). A policy is a mapping from a partial labeling to the next unlabeled example to query. When the active learning policy chooses an unlabeled example, its label according to \( h_{true} \) will be revealed.

A policy can be represented by a policy tree in which each node corresponds to an unlabeled example to query, and edges below a node correspond to its labels. In this paper, we use policy and policy tree interchangeably. A policy can be non-adaptive or adaptive. In a non-adaptive policy, the observed labels are not taken into account when the policy chooses the next example. An adaptive policy, on the other hand, can use the observed labels to determine the next example to query. We will focus on adaptive policies in this paper.

Let \( \Pi_k \) be the set of policy trees of height \( k \). Note that \( \Pi_{|\mathcal{X}|} \) contains full policy trees, while \( \Pi_k \) with \( k < |\mathcal{X}| \) contains partial policy trees. Following the insight in (Cuong et al., 2013), for any (full or partial) policy \( \pi \), we define a probability distribution \( p_0^\pi[\cdot] \) over the paths from the root to a leaf of \( \pi \). Intuitively, \( p_0^\pi[\rho] \) is the probability that the policy \( \pi \) follows the path \( \rho \) during its execution. This probability distribution is induced by the randomness of \( h_{true} \) and is defined as \( p_0^\pi[\rho] \equiv p_0[y_\rho; x_\rho] \), where \( x_\rho \) (resp. \( y_\rho \)) is the sequence of examples (resp. labels) along path \( \rho \). Some objective functions for pool-based active learning can be defined using this probability distribution.

### 3 SUBMODULARITY

Our objective in active learning can often be stated as maximizing some average or worst-case performance with respect to some utility function \( f(S) \) in the non-adaptive case, or \( f(S, h) \) in the adaptive case, where \( S \) is the set of chosen examples. When \( f(S) \) is submodular or \( f(S, h) \) is adaptive submodular, greedy algorithms are known to be near-optimal (Nemhauser et al., 1978; Golovin and Krause, 2011). We shall briefly summarize some results about greedy optimization of submodular functions and adaptive submodular functions, then prove a new result about the worst-case near-optimality of a greedy algorithm for maximizing a pointwise submodular function.\(^2\)

#### 3.1 NEAR-OPTIMALITY OF SUBMODULAR MAXIMIZATION

A set function \( f : 2^\mathcal{X} \to \mathbb{R} \) is submodular if it satisfies the following diminishing return property: for all \( A \subseteq B \subseteq \mathcal{X} \) and \( x \in \mathcal{X} \setminus B \),

\[
f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B).
\]

The function \( f \) is called monotone if \( f(A) \leq f(B) \) for all \( A \subseteq B \).

To select a set of size \( k \) that maximizes a monotone submodular function, one greedy strategy is to iteratively select the next example \( x^* \) that satisfies

\[
x^* = \arg \max_x \{f(S \cup \{x\}) - f(S)\},
\]

where \( S \) is the previously selected examples. The following theorem by Nemhauser et al. (1978) states the near-optimality of this greedy algorithm when maximizing a monotone submodular function.

**Theorem 1** (Nemhauser et al. 1978). Let \( f \) be a monotone submodular function such that \( f(\emptyset) = 0 \), and let \( S_k \) be the set of examples selected up to iteration \( k \) using the greedy criterion in Equation (1). Then for all \( k > 0 \), we have \( f(S_k) \geq (1 - 1/e) \max_{|S|=k} f(S) \).

#### 3.2 NEAR-OPTIMALITY OF ADAPTIVE SUBMODULAR MAXIMIZATION

Adaptive submodularity (Golovin and Krause, 2011) is an extension of submodularity to the adaptive setting. For a partial labeling \( D \) and a full labeling \( h \), we write \( h \sim D \) to

\(^1\) In our setting, the usual objective of determining the true hypothesis \( h_{true} \) is infeasible unless the support of \( p_0 \) is small. When \( p_0[h] > 0 \) for all \( h \), we need to query the whole pool \( \mathcal{X} \) in order to determine \( h_{true} \).

\(^2\) Note that our result can also be applied to settings other than active learning.
denote that $D$ is consistent with $h$. That is, $D \subseteq h$ when we view a labeling as a set of $(x, y)$ pairs. For two partial labelings $D$ and $D'$, we call $D$ a sub-labeling of $D'$, if $D \subseteq D'$.

We consider a utility function $f : 2^X \times Y^X \to \mathbb{R}_{\geq 0}$ which depends on the examples selected and the true labeling of $X$. For a partial labeling $D$ and an example $x$, we define $\Delta(x|D) \triangleq \mathbb{E}_h[f(\text{dom}(D) \cup \{x\}, h) - f(\text{dom}(D), h)]$, where the expectation is with respect to $p_0[h | h \sim D]$ and $\text{dom}(D)$ is the domain of $D$.

From the definitions in (Golovin and Krause, 2011), $f$ is adaptive submodular with respect to $p_0$ if for all $D$ and $D'$ such that $D \subseteq D'$, and for all $x \in X \setminus \text{dom}(D')$, we have $\Delta(x|D) \geq \Delta(x|D')$. Furthermore, $f$ is adaptive monotone with respect to $p_0$ if for all $D$ with $p_0[h \sim D] > 0$ and for all $x \in X$, we have $\Delta(x|D) \geq 0$.

Let $\pi$ be a policy for selecting the examples and $x^*_h$ be the set of examples selected by $\pi$ under the true labeling $h$. We define the expected utility of $\pi$ as $f_{\text{avg}}(\pi) \triangleq \mathbb{E}[f(x^*_h, h)]$, where the expectation is with respect to $p_0[h]$. To adaptively select a set of size $k$ that maximizes $f_{\text{avg}}$, one greedy strategy is to iteratively select the next example $x^*$ that satisfies

$$x^* = \arg \max_x \Delta(x|D), \quad (2)$$

where $D$ is the partial labeling that has already been observed. The following theorem by Golovin and Krause (2011) states the near-optimality of this greedy policy when $f$ is adaptive monotone submodular.

**Theorem 2** (Golovin and Krause 2011). Let $f$ be an adaptive monotone submodular function with respect to $p_0$, $\pi$ be the adaptive policy selecting $k$ examples using Equation (2), and $\pi^*$ be the optimal policy with respect to $f_{\text{avg}}$ that selects $k$ examples. Then for all $k > 0$, we have $f_{\text{avg}}(\pi) > (1 - 1/e) f_{\text{avg}}(\pi^*)$.

### 3.3 NEAR-OPTIMALITY OF POINTWISE SUBMODULAR MAXIMIZATION

Theorem 2 gives near-optimal average-case performance guarantee for greedily optimizing an adaptive monotone submodular function. We now give a new near-optimal worst-case performance guarantee for greedily optimizing a pointwise monotone submodular function. A utility function $f : 2^X \times Y^X \to \mathbb{R}_{\geq 0}$ is said to be pointwise submodular if the set function $f_h(S) \triangleq f(S, h)$ is submodular for all $h$. Similarly, $f$ is pointwise monotone if $f_h(S)$ is monotone for all $h$.

When $f$ is pointwise monotone submodular, the average utility $f_{\text{avg}}(S) = \mathbb{E}_{h \sim p_0}[f(S, h)]$ is monotone submodular, and thus the non-adaptive greedy algorithm is a near-optimal non-adaptive policy for maximizing $f_{\text{avg}}(S)$ (Golovin and Krause, 2011). However, we are more interested in the adaptive policies in this section.

For any partial labeling $D$, any $x \in X \setminus \text{dom}(D)$, and any $y \in Y$, we write $D \cup \{(x, y)\}$ to denote the partial labeling $D$ with an additional mapping from $x$ to $y$.

We assume that for any $S \subseteq X$ and any labeling $h$, the value of $f(S, h)$ does not depend on the labels of examples in $X \setminus S$. We call this the minimal dependency property. Let us extend the definition of $f$ so that its second parameter can be a partial labeling. The minimal dependency property implies that for any partial labeling $D$ and any labeling $h \sim D$, we have $f(\text{dom}(D), h) = f(\text{dom}(D), D)$. Without this minimal dependency property, the theorem in this section may not hold. We will see some examples of functions that satisfy or do not satisfy the minimal dependency property in Section 4 and 5.

For a partial labeling $D$ and an example $x$, define

$$\delta(x|D) \triangleq \min_{y \in Y} \{f(\text{dom}(D) \cup \{x\}, D \cup \{(x, y)\}) - f(\text{dom}(D), D)\}.$$  

We consider the adaptive greedy strategy that iteratively selects the next example $x^*$ satisfying

$$x^* = \arg \max_x \delta(x|D), \quad (3)$$

where $D$ is the partial labeling that has already been observed. For any policy $\pi$, let $f_{\text{worst}}(\pi) \triangleq \min_h f(x^*_h, h)$ be the worst-case objective function. The following theorem states the near-optimality of the above greedy policy with respect to $f_{\text{worst}}$ when $f$ is pointwise monotone submodular.

**Theorem 3**. Let $f$ be a pointwise monotone submodular function such that $f(\emptyset, h) = 0$ for all $h$, and $f$ satisfies the minimal dependency property. Let $\pi$ be the adaptive policy selecting $k$ examples using Equation (3), and $\pi^*$ be the optimal policy with respect to $f_{\text{worst}}$ that selects $k$ examples. Then for all $k > 0$, we have $f_{\text{worst}}(\pi) > (1 - 1/e) f_{\text{worst}}(\pi^*)$.

The main idea in proving this theorem is to show that at every step, the greedy policy can cover at least $(1/k)$-fraction of the optimal remaining utility. This property can be proven by replacing the current greedy step with the optimal policy and considering the adversary’s path for this optimal policy. See Appendix A for a proof of this theorem.

We note that in the worst-case setting, Golovin and Krause (2011) also considered the problem of minimizing the number of queries needed to achieve a target utility value. However, their results mainly rely on the condition that the

\footnote{Note that in the definition of $f_{\text{worst}}(\pi)$, $h$ has to range over the set $Y^X$ of all possible labelings. Otherwise, Theorem 3 does not necessarily hold.}
utility function is adaptive submodular, not the pointwise
submodular condition considered in this section. It is also
worth noting that our new greedy criterion in Equation (3)
is different from the greedy criterion considered by Golovin
and Krause (2011), which is essentially Equation (2). Thus,
our result does not follow from their result and is developed
using a different argument.

4 PROPERTIES OF GREEDY ACTIVE
LEARNING CRITERIA

We now briefly introduce three greedy criteria that have
been used for active learning: maximum entropy, maxi-
mum Gibbs error, and least confidence. These criteria are
equivalent in the binary-class case (i.e. they all choose the
same examples to query), but they are different in the multi-
class case. We will prove some new properties of the max-
imum entropy and the least confidence criteria.

4.1 MAXIMUM ENTROPY

The maximum entropy criterion chooses the next exam-
ple whose posterior label distribution has the maximum
Shannon entropy (Settles, 2010). Formally, this criterion
chooses the next example \( x^* \) that satisfies

\[
x^* = \arg \max_x E_{y \sim p_D[y|x]} [- \ln p_D[y|x]], \tag{4}
\]

where \( p_D \) is the posterior obtained after observing the par-
tial labeling \( D \). From (Cuong et al., 2013), it is desirable to
maximize the policy entropy

\[
H_{\text{ent}}(\pi) \equiv \mathbb{E}_{p \sim p^*_D} [- \ln p^*_0[\rho]],
\]

where the expectation is over all the paths in the policy tree
of \( \pi \), as maximizing the policy entropy will minimize the
expected label entropy given the observations. Criterion
(4) can be viewed as a greedy algorithm for maximizing the
policy entropy.

Due to the monotonicity and submodularity of Shannon en-
tropy (Fujishige, 1978), we can construct a non-adaptive
greedy policy that achieves near-optimality with respect to
the objective function \( H_{\text{ent}} \) in the non-adaptive setting. In
the adaptive setting, however, it was previously unknown
whether the maximum entropy criterion is near-optimal with
respect to \( H_{\text{ent}} \) (Cuong et al., 2013).

We now show that, in general, the maximum entropy crite-
ron may not be near-optimal with respect to the objective function \( H_{\text{ent}} \) (Theorem 4).

**Theorem 4.** Let \( \pi \) be the adaptive policy in \( \Pi_k \) selecting
examples using Equation (4), and \( \pi^* \) be the optimal ad-
pative policy in \( \Pi_k \) with respect to \( H_{\text{ent}} \). For any \( 0 < \alpha < 1 \),
there exists a problem where \( H_{\text{ent}}(\pi) / H_{\text{ent}}(\pi^*) < \alpha. \)

The main idea in proving this theorem is to construct a set
of independent distractor examples that have highest en-
tropy but provide no information about the true hypothe-
sis. The greedy criterion is tricked to choose only these
distractor examples. On the other hand, there is an identi-
fer example which gives the identity of the true hypothesis
but has a lower entropy than the distractor examples. Once
the label of the identifier example is revealed, there will be
a number of high entropy examples to query, so that the
policy entropy achieved is higher than that of the greedy
algorithm. See the supplement for a proof of this theorem.

4.2 MAXIMUM GIBBS ERROR

The maximum Gibbs error criterion chooses the next ex-
ample whose posterior label distribution has the maximum
Gibbs error (Cuong et al., 2013). Formally, this criterion
chooses the next example \( x^* \) that satisfies

\[
x^* = \arg \max_x E_{y \sim p_D[y|x]} [1 - p_D[y|x]]. \tag{5}
\]

This criterion attempts to greedily maximize the policy
Gibbs error

\[
H_{\text{gibbs}}(\pi) \equiv \mathbb{E}_{p \sim p^*_D} [1 - p^*_0[\rho]],
\]

which is a lower-bound of the policy entropy \( H_{\text{ent}}(\pi) \).

It has been shown by Cuong et al. (2013, sup.) that the pol-
icy Gibbs error \( H_{\text{gibbs}} \) corresponds to the expected version
space reduction in \( \mathcal{H} \). Furthermore, the maximum Gibbs
error criterion in Equation (5) corresponds to the algorithm
that greedily maximizes the expected version space reduc-
tion. For \( S \subseteq \mathcal{X} \) and \( h \in \mathcal{H} \), the version space reduction
function is defined as \( f(S, h) \equiv 1 - p_0[h(S); S] \).
Since the version space reduction function is adaptive
monotone submodular (Golovin and Krause, 2011), the
maximum Gibbs error criterion is near-optimal with respect
to the objective function \( H_{\text{gibbs}} \) in both the non-adaptive
and adaptive settings. That is, the greedy policy using Equation
(5) has the policy Gibbs error within a factor \((1 - 1/e)\) of
the optimal policy (Cuong et al., 2013).

4.3 LEAST CONFIDENCE

The least confidence criterion chooses the next example
whose most likely label has minimal posterior probabili-
ty (Lewis and Gale, 1994; Culotta and McCallum, 2005).
Formally, this criterion chooses the next examples \( x^* \) that
satisfies

\[
x^* = \arg \min_x \{ \max_{y \in Y} p_D[y|x] \}. \tag{6}
\]

Note that \( x^* = \arg \max_x \{ 1 - \max_{y \in Y} p_D[y|x] \} \). Thus,
the least confidence criterion greedily optimizes the error
rate of the Bayes classifier on the distribution \( p_D[\cdot|x] \). In
this section, we use the result in Section 3.3 to prove that
the least confidence criterion near-optimally maximizes the worst-case version space reduction.

For a policy $\pi$, we define the worst-case version space reduction objective as

$$H_L(\pi) \triangleq \min_h f(x^*_h, h)$$

where $f$ is the version space reduction function defined in Section 4.2. We note that $f$ satisfies the minimal dependency property. It can also be shown that $f$ is pointwise monotone submodular, and the least confidence criterion is equivalent to the criterion in Equation (3). Thus, it follows from Theorem 3 that the least confidence criterion is near-optimal with respect to the objective function $H_L$ (Theorem 5). See the supplement for a proof.

**Theorem 5.** Let $\pi$ be the adaptive policy in $\Pi_k$ selecting examples using Equation (6), and $\pi^*$ be the optimal adaptive policy in $\Pi_k$ with respect to $H_L$. For all $k > 0$, we have $H_L(\pi) > (1 - 1/e)H_L(\pi^*)$.

## 5 ACTIVE LEARNING WITH GENERAL LOSS

In this section, let us focus on the maximum Gibbs error criterion in Section 4.2. The policy Gibbs error objective $H_{\text{gibbs}}$ can be written as $H_{\text{gibbs}}(\pi) = E_{h \sim p_0}[f(x^*_h, h)]$, where $f$ is the version space reduction function (Cuong et al., 2013, sup.). Note that $f(x^*_h, h)$ is the expected 0-1 loss that a random labeling drawn from $p_0$ differs from $h$ on $x^*_h$. Because of the nature of 0-1 loss, even if the random labeling only differs from $h$ on one element of $x^*_h$, it is counted as an error.

To overcome this disadvantage, we formulate a new objective function that can handle an arbitrary general loss function $L : \mathcal{Y}^X \times \mathcal{Y}^X \rightarrow \mathbb{R}_{\geq 0}$ satisfying the following two properties: $L(h, h') = L(h', h)$ for any two labelings $h$ and $h'$ of $\mathcal{X}$, and if $h = h'$ then $L(h, h') = 0$. For $S \subseteq \mathcal{X}$ and $h \in \mathcal{H}$, we define the generalized version space reduction function

$$f_L(S, h) \triangleq E_{h' \sim p_0}[L(h, h') 1(h(S) \neq h'(S))].$$

Note that $f_L(S, h) = \sum_{h' : h(S) \neq h'(S)} p_0[h']L(h, h')$, which can be written as

$$\sum_{h'} p_0[h']L(h, h') - \sum_{h' : h(S) = h'(S)} p_0[h']L(h, h').$$

If $L$ is the 0-1 loss, i.e., $L(h, h') = 1(h \neq h')$, we have $f_{0,1}(S, h) = \sum_{h' : h(S) \neq h'(S)} p_0[h']$, which is equal to the version space reduction function $f(S, h)$.

Our new objective is to maximize the expected value of the generalized version space reduction

$$H_L^{\text{avg}}(\pi) \triangleq E_{h \sim p_0}[f_L(x^*_h, h)].$$

When $L$ is the 0-1 loss, this objective function is equal to the policy Gibbs error $H_{\text{gibbs}}(\pi)$. Thus, we call $H_L^{\text{avg}}(\pi)$ the generalized policy Gibbs error.

### 5.1 AVERAGE-CASE CRITERION

To maximize $H_L^{\text{avg}}(\pi)$, a natural algorithm is to greedily maximize $f_L$ at each step. Let $D$ be the previously observed partial labeling, this greedy criterion chooses the next example $x^*$ that satisfies

$$x^* = \arg \max_x E_{h \sim p_D}[f_L(\{x\} \cup \{x\}, h) - f_L(\{x\}, h)]$$

We call this criterion the average generalized Gibbs error criterion.

From the result in Section 3.2, if $f_L$ is adaptive monotone submodular, then using the average generalized Gibbs error criterion is near-optimal. Theorem 6 below states this result, which is a direct consequence of Theorem 2.

**Theorem 6.** Let $\pi_L^{\text{avg}}$ be the adaptive policy in $\Pi_k$ selecting examples using Equation (7), and $\pi^*$ be the optimal adaptive policy in $\Pi_k$ with respect to $H_L^{\text{avg}}$. If $f_L$ is adaptive monotone submodular with respect to the prior $p_0$, then $H_L^{\text{avg}}(\pi_L^{\text{avg}}) > (1 - 1/e)H_L^{\text{avg}}(\pi^*)$.

Note that if $L$ is 0-1 loss, then $f_L$ is adaptive monotone submodular with respect to any prior. Unfortunately, in general, $f_L$ may not be adaptive submodular with respect to a prior $p_0$ (Theorem 7). See the supplement for a proof.

**Theorem 7.** Let $p_0$ be a prior with $p_0[h] > 0$ for all $h$. There exists a loss function $L$ such that $f_L$ is not adaptive submodular with respect to $p_0$.

In the supplementary material, we also discuss a sufficient condition for $f_L$ to be adaptive monotone submodular with respect to $p_0$, and hence satisfy the precondition in Theorem 6. However, it remains open whether this sufficient condition is true for any interesting loss function other than 0-1 loss.

### 5.2 WORST-CASE CRITERION

We have shown in Theorem 7 that $f_L$ may not be adaptive submodular, and thus we may not always have a theoretical guarantee for the average generalized Gibbs error criterion. In this section, we will reconsider our objective in the worst case instead of the average case.

In the worst case, we may want to maximize the objective function $H_L^{\text{worst}}(\pi) \triangleq \min_h f_L(x^*_h, h)$. However, using this objective function may be too conservative since the generalized version space reduction is computed only from the losses between the surviving labelings and the worst-
case labeling. Instead, we propose a less conservative objective function based on the losses among all the surviving labelings. Formally, we define the following total generalized version space reduction function

\[ t_L(S, h) \equiv \sum_{h' \in H} \sum_{h'' \in H} p_0(h') L(h', h'') p_0(h'') \]

\[ - \sum_{h': h'(S) = h(S)} \sum_{h'': h''(S) = h(S)} p_0(h') L(h', h'') p_0(h''). \]

Our new objective is to maximize the following function called the worst-case total generalized policy Gibbs error

\[ T^\text{worst}_L(\pi) \equiv \min_h t_L(x^*_h, h). \]

To maximize \( T^\text{worst}_L \), we propose a greedy algorithm that maximizes the worst-case total generalized version space reduction at each step. Note that \( t_L(S, h) \) satisfies the minimal dependency property, i.e., its value does not depend on the labels of \( X \setminus S \) in \( h \). So, for a partial labeling \( D \), we have \( t_L(\text{dom}(D), h) = t_L(\text{dom}(D), D) \) for any \( h \sim D \). Using this notation, the greedy criterion for choosing the next example \( x^* \) can be written as

\[ x^* = \arg \max_x \min_{y \in Y} \{ t_L(\text{dom}(D) \cup \{x\}, D \cup \{(x, y)\}) \} \]

\[ -t_L(\text{dom}(D), D) \} \right \} \] (8)

where \( D \) is the previously observed partial labeling. We call this criterion the worst-case generalized Gibbs error criterion. It can be shown that \( t_L \) is pointwise monotone submodular and satisfies the minimal dependency property for any loss function \( L \). Furthermore, the criterion in Equation (8) is equivalent to the criterion in Equation (3). Thus, it follows from Theorem 3 that this greedy criterion is near-optimal with respect to the objective function \( T^\text{worst}_L(\pi) \) (Theorem 8). See the supplement for a proof.

**Theorem 8.** Let \( \pi^\text{worst}_L \) be the adaptive policy in \( \Pi_k \) selecting examples using Equation (8), and \( \pi^* \) be the optimal adaptive policy in \( \Pi_k \) with respect to \( T^\text{worst}_L \). We have \( T^\text{worst}_L(\pi^\text{worst}_L) > (1 - 1/e) T^\text{worst}_L(\pi^*) \).

It is worth noting that, like \( t_L \), the function \( f_L \) is also pointwise submodular for any loss function \( L \). The proof for the pointwise submodularity of \( f_L \) is essentially similar to the proofs that \( f \) and \( t_L \) are pointwise submodular in Theorem 5 and Theorem 8 (see the supplement for a proof of this claim). However, \( f_L \) does not satisfy the minimal dependency property. Besides, Theorem 7 also shows that \( f_L \) may not be adaptive submodular. Thus, this is an example that a pointwise submodular function is not necessarily adaptive submodular, and we may not be able to use Golovin and Krause (2011)’s result to obtain a result in the average case for pointwise submodular functions.

### 5.3 Computing the Criteria

In this section, we discuss the computations of the criteria in Equation (7) and Equation (8). First, we give two propositions below regarding these equations. See the supplement for proofs.

**Proposition 1.** The selected example \( x^* \) in Equation (7) is equal to

\[ \arg \min_x \sum_y \mathbb{E}_{h, h' \sim \mathcal{PD}} [L(h, h') \mathbf{1}(h(x) = h'(x) = y)]. \]

**Proposition 2.** The selected example \( x^* \) in Equation (8) is equal to

\[ \arg \min_x \{ \max_y \mathbb{E}_{h, h' \sim \mathcal{PD}} [L(h, h') \mathbf{1}(h(x) = h'(x) = y)] \}. \]

From these two propositions, we can compute Equation (7) and Equation (8) by estimating the expectation \( \mathbb{E}_{h, h' \sim \mathcal{PD}} [L(h, h') \mathbf{1}(h(x) = h'(x) = y)] \) for each \( y \in Y \). This estimation can be done by sampling from the posterior.

We can sample directly from \( \mathcal{PD} \) two sets \( H \) and \( H' \) which contain samples of \( h \) and \( h' \) respectively. Then, the expectation \( \mathbb{E}_{h, h' \sim \mathcal{PD}} [L(h, h') \mathbf{1}(h(x) = h'(x) = y)] \) can be approximated by

\[ \frac{1}{|H| \times |H'|} \sum_{h \in H} \sum_{h' \in H'} L(h, h') \mathbf{1}(h(x) = h'(x) = y). \]

Note that this approximation only requires samples of the labelings from the posterior, and we do not need to explicitly maintain the set of all labelings which may be exponentially large. In the case when the labelings are generated by probabilistic models following some prior distribution, sampling from \( \mathcal{PD} \) may be difficult. A simple approximation is to sample \( H \) and \( H' \) from the MAP model.

### 6 Experiments

Experimental results comparing the maximum entropy criterion, the maximum Gibbs error criterion, and the least confidence criterion were reported in (Cuong et al., 2013). In this section, we only focus on the active learning criteria with general loss functions, and conduct experiments with two common loss functions used in practice: the Hamming loss and the \( F_1 \) loss. For two labelings \( h \) and \( h' \) (viewing them as label vectors), the Hamming loss is the Hamming distance between them, and the \( F_1 \) loss is \( 1 - F_1(h, h') \) where \( F_1(h, h') \in [0, 1] \) is the \( F_1 \) score between \( h \) and \( h' \).

We experiment with various binary-class tasks from the UCI repository (Bache and Lichman, 2013) and the 20Newsgroups dataset (Joachims, 1996). We use the binary-class logistic regression as our model, and compare the active learners using the greedy criteria in Section 5.1 and 5.2 with the passive learner (Pass) and the maximum Gibbs error active learner (Gibbs). The maximum Gibbs error criterion is estimated from Equation (5) using the MAP
hypothesis. Note that the maximum Gibbs error criterion is equivalent to the maximum entropy and the least confidence criterion in this case since the tasks are binary-class.

We estimate the average-case criteria (AvgH and AvgF) in Section 5.1 and the worst-case criteria (WorstH and WorstF) in Section 5.2 using the approximation in Section 5.3 with the MAP hypothesis. AvgH and WorstH use the Hamming loss, while AvgF and WorstF use the $F_1$ loss. We compare the AUCs (area under the curve) for the accuracy scores of Pass, Gibbs, AvgH, and WorstH. We also compare the AUCs for the $F_1$ scores of Pass, Gibbs, AvgF, and WorstF.

The AUCs are computed from the first 150 examples and normalized so that their ranges are from 0 to 100. We randomly choose the first 10 examples as a seed set. We use the same seed set for all the algorithms.

The detailed procedure to compute the AUCs for our experiments is as follows. We sequentially choose 10 (seed size), 11, ..., 150 training examples using active learning or passive learning. Then for each training size, we train a model and compute its score (accuracy or $F_1$) on a separate test set. Using these scores, we can compute the AUCs. We use the AUC scores because we want to compare the whole learning curves from choosing 10 to 150 training examples, not just the scores at any single point (e.g., 150 examples). This is consistent with previous works such as (Settles and Craven, 2008) and (Cuong et al., 2013).

The results for the UCI datasets are given in Table 2. From Table 2, all the active learning algorithms perform better than passive learning in terms of accuracy. On average, WorstH and AvgH perform slightly better than Gibbs, and WorstH achieves the best average AUC for accuracy. In addition, all the active learning algorithms also perform better than passive learning in terms of $F_1$ score. WorstF and AvgF are also slightly better than Gibbs, and AvgF has the best average AUC for $F_1$ score.

The results for the 20Newsgroups dataset are given in Table 3. From Table 3, all the active learning algorithms are better than passive learning in terms of accuracy. WorstH and AvgH perform slightly better than Gibbs, and WorstH achieves the best average AUC for accuracy. In addition, all the active learning algorithms also perform better than passive learning in terms of $F_1$ score. On average, WorstF and AvgF also perform slightly better than Gibbs, and AvgF has the best average AUC for $F_1$ score.
In both datasets, using the Hamming loss or $F_1$ loss is better than using the 0-1 loss (the Gibbs criterion). Furthermore, the worst-case criterion with Hamming loss achieves the best average scores in terms of accuracy, while the average-case criterion with $F_1$ loss achieves the best average scores in terms of $F_1$.

7 CONCLUSION

We have discussed several theoretical properties of greedy algorithms for active learning. In particular, we proved a negative result for the maximum entropy criterion and a near-optimality result for the least confidence criterion in the worst case. We also considered active learning with general losses and proposed two greedy algorithms, one of which is for the average case and the other is for the worst case. Our experiments show that the new algorithms perform well in practice.

A APPENDIX: PROOF OF THEOREM 3

Let $\pi$ and $\pi^*$ be the policies as in the statement of Theorem 3. Let $h_\pi = \arg\min_h f(x^\pi_n, h)$. Then we have $f_{\text{worst}}(\pi) = f(x^*_{h_\pi}, h_\pi)$. Note that $h_\pi$ corresponds to a path from the root to a leaf of the policy tree of $\pi$. Let the examples and labels along the path $h_\pi$ (from the root of the tree to a leaf) be: $h_\pi = \{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$.

Since $f$ satisfies the minimal dependency property, let us abuse the notation and write $f(\{x_t\}_{t=1}^i, \{y_t\}_{t=1}^i)$ to denote $f(\{x_t\}_{t=1}^i, \{y_t\}_{t=1}^i)$.

$$u_t \eqdef f(\{x_t\}_{t=1}^i, \{y_t\}_{t=1}^i) - f(\{x_t\}_{t=1}^{i-1}, \{y_t\}_{t=1}^{i-1})$$

$$v_i \eqdef \sum_{t=1}^i u_t$$

and $z_i \eqdef f_{\text{worst}}(\pi^*) - v_i$.

We prove the following claims.

Claim 1. For all $i$, we have $u_{i+1} \geq z_i/k$.

Proof. Consider the case that after observing $(x_1, y_1), \ldots, (x_i, y_i)$, we run the policy $\pi^*$ from its root and only follow the paths consistent with $(x_1, y_1), \ldots, (x_i, y_i)$ down to a leaf. In this case, all the paths of the policy $\pi^*$ must obtain a value at least $z_i = f_{\text{worst}}(\pi^*) - v_i$, because running $\pi^*$ without any observation would obtain at least $f_{\text{worst}}(\pi^*)$ and the observations $(x_1, y_1), \ldots, (x_i, y_i)$ cover a value $v_i$.

Now we consider the adversary’s path of the policy $\pi^*$ in this scenario which is defined as

$$h_{\text{adv}} \eqdef \{(x_{1, \text{adv}}, y_{1, \text{adv}}), (x_{2, \text{adv}}, y_{2, \text{adv}}), \ldots, (x_{k, \text{adv}}, y_{k, \text{adv}})\},$$

where $y_{\text{adv}} = \arg\min_y \{f(\{x_t\}_{t=1}^i \cup \{x_{t, \text{adv}}\}_{t=1}^i \cup \{x_{t, \text{adv}}\}_{t=1}^j - 1, \{y_t\}_{t=1}^i \cup \{y_{t, \text{adv}}\}_{t=1}^j - 1)

- f(\{x_t\}_{t=1}^i \cup \{x_{t, \text{adv}}\}_{t=1}^i - 1, \{y_t\}_{t=1}^i \cup \{y_{t, \text{adv}}\}_{t=1}^j - 1)}$$

if $x_{\text{adv}}$ has not appeared in $\{x_1, \ldots, x_i\}$. Otherwise, if $x_{\text{adv}} = x_t$ for some $t \in \{1, \ldots, i\}$, then $y_{\text{adv}} = y_t$. From the previous discussion, $h_{\text{adv}}$ covers a value of at least $z_i$ in $k$ steps. Thus, one of its steps must cover a value of at least $z_i/k$.

Hence, what remains is to show that doing the greedy step in $\pi$ after observing $(x_1, y_1), \ldots, (x_i, y_i)$ is better than any single step along $h_{\text{adv}}$. In the trivial case where $(x_{\text{adv}}, y_{\text{adv}}) \in \{(x_1, y_1), \ldots, (x_i, y_i)\}$, we obtain nothing in this step since $(x_{\text{adv}}, y_{\text{adv}})$ has already been observed. Thus, the above is true in this case. In the non-trivial case,

$$\frac{u_{i+1}}{k} = f(\{x_t\}_{t=1}^{i+1}, \{y_t\}_{t=1}^{i+1}) - f(\{x_t\}_{t=1}^i, \{y_t\}_{t=1}^i) \geq \min_y \{f(\{x_t\}_{t=1}^i \cup \{x_{t, \text{adv}}\}_{t=1}^i \cup \{y_t\}_{t=1}^i \cup \{y_{t, \text{adv}}\}_{t=1}^i - 1)\}

- f(\{x_t\}_{t=1}^i, \{y_t\}_{t=1}^i)

\geq \min_y \{f(\{x_t\}_{t=1}^i \cup \{x_{t, \text{adv}}\}_{t=1}^i - 1 \cup \{x_{t, \text{adv}}\}_{t=1}^j, \{y_t\}_{t=1}^i \cup \{y_{t, \text{adv}}\}_{t=1}^j - 1)\}

- f(\{x_t\}_{t=1}^i \cup \{x_{t, \text{adv}}\}_{t=1}^i - 1, \{y_t\}_{t=1}^i \cup \{y_{t, \text{adv}}\}_{t=1}^j - 1).

Note that the second inequality is due to the greedy criterion, and the third inequality is due to the submodularity of $f$ on the adversary path. Therefore, this claim is true.

Claim 2. For all $i \geq 0$, we have $z_i \leq (1 - \frac{1}{k}) f_{\text{worst}}(\pi^*)$.

Proof. We prove this claim by induction. For $i = 0$, this holds because $z_0 = f_{\text{worst}}(\pi^*)$ by definition. Assume that $z_i \leq (1 - \frac{1}{k}) f_{\text{worst}}(\pi^*)$, then due to Claim 1,

$$z_{i+1} = f_{\text{worst}}(\pi^*) - v_{i+1} = f_{\text{worst}}(\pi^*) - v_i - u_{i+1} \leq z_i - u_{i+1} \leq z_i - \frac{z_i}{k} = \left(1 - \frac{1}{k}\right) z_i \leq \left(1 - \frac{1}{k}\right) f_{\text{worst}}(\pi^*)$.$$

Therefore, this claim is true.

To prove Theorem 3, we apply Claim 2 with $i = k$ and have $z_k \leq (1 - \frac{1}{k}) f_{\text{worst}}(\pi^*) < \frac{1}{k} f_{\text{worst}}(\pi^*)$. Hence, $f_{\text{worst}}(\pi) = v_k = f_{\text{worst}}(\pi^*) - z_k > (1 - \frac{1}{k}) f_{\text{worst}}(\pi^*)$.

Acknowledgements

This work is supported by the US Air Force Research Laboratory under agreement number FA2386-12-1-4031.
References


