Selective Sampling on Probabilistic Data

Peng Peng* Raymond Chi-Wing Wong*

Abstract

In the literature of supervised learning, most existing studies assume that the labels provided by the labelers are *deterministic*, which may introduce noise easily in many real-world applications. In many applications like crowdsourcing, however, many labelers may simultaneously label the same group of instances and thus the label of each instance is associated with a probability. Motivated by this observation, we propose a new framework where each label is enriched with a probability. In this paper, we study an interactive sampling strategy, namely, *selective sampling*, in which each selected instance is labeled with a probability.Specifically, we flip a coin every time when we read a new instance and decide whether it should be labeled according to the flipping result. We prove that in our setting the label complexity can be reduced dramatically. Finally, we conducted comprehensive experiments in order to verify the effectiveness of our proposed labeling framework.

1 Introduction

Selective Sampling[16] has been studied extensively in the literature. In most circumstances, when unlabeled instances are read sequentially, selective sampling allows us to save the cost of labeling in supervised learning and learn a classifier which is as good as those learned with many more labeled instances. We aim at showing that the labeling cost can be further reduced when the classifier is learned based on the training dataset with probabilistic labels.

Consider a binary classification problem. Traditionally, the training dataset used for supervised learning contains *deterministic labels* only. Given an instance, we say that its label is *deterministic* if this instance has a label exactly equal to either 0 or 1. However, we can use *probabilistic labels* for learning a classifier as well. That is, instead of labeling an instance with a 0-1 label, we could label an instance with a probability value which denotes the probability that its label is 1. In many cases, probabilistic labels can be obtained easily. In crowdsourcing applications where each label is given by multiple labelers [12], the proportion of labelers giving a particular label corresponds to the probability that the instance has this label. In the medical diagnosis application, a patient's disease is diagnosed as Coronary Heart Disease with 50%-60% probability by a doctor based on the result of an electrocardigogram (ECG) test [25]. In the galaxy image recognition application, an existence of a volcano is determined with 60% probability by an astro-researcher based on the observation that "the surface shown in the image contains no summit visible but with evidence of flanks or circular outline" [27]. In natural language processing, due to multiple meanings of some frequent words (e.g., 7.8 meanings on average for each of the 121 frequent words) [1], some existing studies [21] which make use of *word sense disambiguation (WSD)* to reduce some possible meanings of a word give the 75.2% accuracy for estimating the meaning of a word.

Probabilistic Labels: In this paper, we propose to learn a classifier from an "enriched" version of labels called *probabilistic fractional scores* (or *fractional scores* in short). Consider a binary classification with two classes, class 0 and class 1. Each instance is associated with a fractional score (instead of 0/1 labels) which denotes the probability that this instance belongs to class 1. We consider an active learning algorithm which delicately selects an "informative" instance for labeling in each round in terms of a probability which is related to the corresponding fractional score and finds an accurate classifier based on these selected "informative" instances.

There are two major challenges for designing a supervised learning algorithm. The first challenge is whether a *tight* theoretical bound on the *label complexity*, which is defined to be the minimum number of instances used for classification, can be derived for an accurate algorithm/classifier whose *error* is at most an error parameter ϵ where $\epsilon \in [0, 1]$. Most existing studies [4, 13, 18, 5] focus on finding this theoretical bound in either a *realizable* setting or an *agnostic* setting. Both settings are defined in terms of the distribution which generate the label of a given instance. The realizable setting corresponds to such a group of distributions that always return the same label for a given instance, which is quite optimistic, while the agnostic setting corresponds to a group of distributions which may generate the label of a given instance arbitrarily, which is quite pessimistic. We say that there is no noise in the realizable setting, while it could be very noisy in the agnostic setting. However, in many real-life applications, we obtain a dataset sampled from a distribution which contains *well-behaved* noise. In [30], the authors found that the well-behaved noise can be described by the *margin* assumption (which will be discussed in this paper). Under this assumption, there is a distribution-dependent parameter γ which is a non-negative real number

*The Hong Kong University of Science and Technology

used to describe the well-behaved noise. Due to its advantage to tighten the theoretical bound, This assumption has been commonly adopted in a lot of studies [28, 26, 6, 19]. For example, [28] considering this assumption showed a *fast* rate of the *convergence* of an SVM model. Some other examples considering this assumption include the decision tree [26], the least square regression [3] and the regularized boosting [6]. In this paper, we derive our theoretical bound based on this assumption in active learning.

The second challenge is whether the algorithm is *tractable* and thus can be executed efficiently. Firstly, it is very usual that an algorithm with a very good theoretical bound is intractable. To the best of our knowledge, both the passive learning algorithm [30] and the active learning algorithm [19] considering the margin assumption which have the best-known theoretical bounds are intractable and thus are inefficient. Secondly, it is also usual that an algorithm which is tractable does not give a tight theoretical bound. To the best of our knowledge, there is only one tractable selective sampling strategy (importance weighted active learning [5]) considering the margin assumption but it does not result in a "good" theoretical bound.

In this paper, we consider the importance weighted active learning algorithm learned from a probabilistic dataset such that these two challenges can be addressed. That is, the algorithm has a tight theoretical bound on the label complexity, while it is still tractable. Specifically, the algorithms in [30] and [19], though with good theoretical bounds, are not tractable. Although [5] is tractable, the label complexity of [5] (i.e., $O(\epsilon^{-2})$) is higher than that of our algorithm (i.e., $O(\epsilon^{-\frac{4}{2+\gamma}})$) since $\epsilon^{-2} \geq \epsilon^{-\frac{4}{2+\gamma}}$ for any $\gamma \geq 0$.

It is worth mentioning that a lot of recent studies [10, 22, 24, 31] can use our theoretical bound on the label complexity of the classifier studied in the paper for their label complexity analysis so that their approaches can have solid theoretical guarantees. For example, in [31], the authors proposed a novel SVM model which learns a classifier based on the scenario that only a portion of labels of the instances in the dataset are known. Since the objective function in their optimization algorithm belongs to a regularized empirical risk minimization, which is similar to ours, our technique can also be used to derive the label complexity of their proposed SVM model. The difference between the minimization function in our paper and that in their paper is that other than the model complexity, the regularizer in their objective function also includes the difference between the true label proportion and the estimated label proportion, which is more complicated for the analysis. In [10], the authors investigated the conditions of a proper loss function for estimating the posterior probability from partially labeled data. We can naturally plug their strategy into our algorithm for finding a proper loss, and derive the error bounds based on the discovered proper loss. In [22], the authors studied the problem of estimating labels from label proportions from multiple different perspectives, proposing variant strategies for solving this problem. Even though they have already analyzed the convergence bounds for their probability estimators, our theoretical analysis is rather different from theirs, which may provide a supportive evidence on their theoretical results. In [24], the authors studied the problem of learning from *group probabilities*, which is referred to the posterior probabilities for a subset of the whole dataset. Since they did not provide a theoretical guarantee on their estimated probabilities on the whole dataset, our theoretical analysis can be adaptively used for analyzing their problem as well.

Contributions: There are the following contributions. Firstly, to the best of our knowledge, we are the first to study the active learning algorithm from a training set with probabilistic labels. Secondly, we propose an active sampling strategy by leveraging the probabilistic information in the training dataset. The major idea of the sampling strategy follows the rule of *uncertainty sampling*, but we compute the uncertainty from the fractional scores. Thirdly, we show that there is a theoretical bound on the label complexity of our proposed algorithm. In particular, this result is better than the traditional result in the same setting in most cases. Fourthly, experiments were conducted to show its superior performance over the traditional active learning framework.

The remainder of our paper is organized as follows. We give the problem definition in Section 2. In Section 3, we propose our Fractional Score-based Active Learning (FSAL) algorithm, which outputs a classifier with the help of fractional scores. In Section 4, we show the experimental results. In Section 5, we present the related works. Finally, we conclude our work and discuss some future works in Section 6.

2 Problem Definition

Traditional Setting: Consider a binary classification with two classes 0 and 1. In the *traditional* setting, we are given a training dataset S called a *deterministic dataset* containing n instances, namely $I_1, I_2, ..., I_n$. Each instance I_i is associated with a feature vector and a target attribute where $i = 1, 2, ..., n$. Let $\mathcal{X}$ be the set of all possible feature vectors. Note that there are two possible values, namely 0 and 1, in the target attribute. Given an instance I_i where $i = 1, 2, ..., n$, the content of its feature vector is denoted by $\mathbf{x}_i \in \mathcal{X}$ and the content of its target attribute is denoted by $y_i \in \{0, 1\}$. A *classifier* $h(\cdot)$ is defined to be a *hypothesis* or a function which takes a feature vector $\mathbf{x}$ as an input and outputs either 0 or 1. In the following, for clarity, in some cases, we simply denote $h(\cdot)$ by h.

Following [2, 4, 13, 18], we assume the process of data generation as follows. Let X and Y be the random variables denoting the feature vector and the target attribute of an instance, respectively. All instances are generated according to an underlying joint distribution on two random variables X and Y, denoted by $P(X, Y)$. Given a feature vector $\mathbf{x}$, the conditional probability $P(Y = 1|X = \mathbf{x})$ is denoted by $\eta(\mathbf{x})$. We also assume that the data points in the dataset are *identically independently distributed (i.i.d.)* when the joint distribution is considered.

Given a classifier h, the *expected error* of h, denoted by $err(h)$, is defined to be $P_{(\mathbf{x},y)\sim P(X,Y)}(y \neq h(\mathbf{x}))$. The *Bayes classifier*, denoted by h^*, is defined to be the classifier which gives the minimum expected error. Note that $h^* = \mathbb{I}_{\eta(\mathbf{x})\geq 0.5}$. Given a classifier h, the *excess error* of h is defined to be the difference between its expected error and the expected error of h^*. That is, the *excess error* of h, denoted by $E(h)$, is equal to $err(h) - err(h^*)$. Note that $E(h)$ must be non-negative.

Our Setting: In our setting, we are given a *probabilistic dataset* S_f instead of a *deterministic dataset*. Similar to the traditional setting, S_f contains n instances, namely $I_1, I_2, ..., I_n$. Each instance I_i is associated with a feature vector and a *fractional score* (instead of a target attribute) where $i = 1, 2, ..., n$. This fractional score is a real number ranging from 0 to 1 and corresponds to the likeliness that this instance belong to class 1. If this score is near to 1, then it is likely that this instance belongs to class 1. If it is near to 0, then it is unlikely that this instance belongs to class 1 and instead, it is likely that this instance belongs to class 0. This score can be obtained by labelers and some statistical information (e.g., the medical statistical history) described in Section 1. Given an instance I_i where $i = 1, 2, ..., n$, the content of its feature vector is denoted by $\mathbf{x}_i \in \mathcal{X}$ and the content of its fractional score is denoted by $f_i \in [0, 1]$. Note that if each f_i is equal to either 0 or 1 where $i = 1, 2, ..., n$, then the probabilistic dataset is equivalent to the deterministic dataset.

Consider an instance with its feature x and its fractional score f. Since f is obtained by labelers and some statistical information, it can be regarded as an "observed" version of $\eta(\mathbf{x})$, and thus it may deviate from $\eta(\mathbf{x})$. Following some existing studies [23], we model the deviation with the *Gaussian white noise*. Specifically, the *Gaussian white noise* is represented in form of the Gaussian distribution $\mathcal{N}(0, \sigma^2)$ where σ is a standard deviation of this distribution. With this noise condition, each fractional score f_i in the training dataset follows the distribution of $\mathcal{N}(\eta(\mathbf{x}), \sigma^2)$. Note that it is possible that a value randomly sampled from the distribution $\mathcal{N}(\eta(\mathbf{x}), \sigma^2)$ is out of the range between 0 and 1. In this case, the value can be truncated accordingly. Specifically, if this value is smaller than 0, then it can be assigned to 0. If this value is larger than 1, then it can be assigned to 1. However, in our theoretical analysis, we simply adopt the distribution of $\mathcal{N}(\eta(\mathbf{x}), \sigma^2)$ in order to simplify our theoretical analysis. Considering the truncation method is also possible but yields uninteresting and tedious "boundary" cases.

Problem FSAL: Our problem is called *Fractional Score-based Active Learning (FSAL)* on the probabilistic training dataset as follows. Given a probabilistic training dataset S_f, we want to learn a classifier h such that whenever we see a new instance I which has the information about its feature $\mathbf{x}$ but no information about its fractional score f, we can calculate the *estimated* value of $\eta(\mathbf{x})$, denoted by $\hat{\eta}(\mathbf{x})$, and thus we can determine whether we should obtain the fractional score of instance I from an expert accordingly.

There are three issues to be addressed in this problem. The first issue is how to calculate the estimated value $\hat{\eta}(\mathbf{x})$. The second issue is how to determine whether we should obtain the fractional score of a given instance from an expert accordingly. The third issue is how to design a classifier which sovles problem FSAL.

3 Methodology

As we discussed before, there are three issues to be addressed. In this Section, we give the solutions to these issues in Section 3.1, Section 3.2 and Section 3.3.

3.1 Formulation of Fractional Score Estimation

In this section, we give the formulation of the estimated fractional score by using *Gaussian Process Regression*. The reason why we choose Gaussian Process Regression is that it is a popular nonparametric estimator [23], which can give an accurate estimation via perfectly combining the information from the training dataset and the prior knowledge. In order to simplify our discussion, we just give the simplest version of Gaussian Process Regression as an example to estimate the fractional score. There are many recent studies about more complicated versions of Gaussian Process Regression focusing on the scalability issue with large datasets. Considering the scalability issue with more complicated versions is an orthogonal issue in this paper.

In the following, in order to give an accurate estimation of the fractional score, in addition to the estimated value of $\eta(\mathbf{x})$, we find the *variance* of this estimated value, denoted by $\text{Var}(\mathbf{x})$. As we will see in the algorithm to be shown later, we leverage both the estimated value and its variance to decide whether we should obtain the fractional score of an instance from an expert or not.

We use Gaussian process regression to find $\hat{\eta}(\mathbf{x})$ and $Var(\mathbf{x})$ given a feature vector $\mathbf{x}$.

Consider the first prior stage. The *prior* of the Gaussian process is specified by two components. The first component is the prior mean function, denoted by $m(\cdot)$, which takes the features of an instance as an input and returns a real number between 0 and 1 as an output. The second component is the prior covariance function, denoted by $k(\cdot, \cdot)$, which takes two features as inputs and returns a positive real number as an output. Formally, the prior of the Gaussian Process is represented in form of $\mathcal{GP}(m(\cdot), k(\cdot, \cdot))$.

In this stage, we need to set $m(\cdot)$ and $k(\cdot, \cdot)$. We set $m(\cdot)$ to 0.5, because 0.5 corresponds to a random guess when we have no prior knowledge. In the following, we are studying the Gaussian Process in form of $\mathcal{GP}(0.5, k(\cdot, \cdot))$.

We adopt the Radial Basis Function (RBF), one of the most commonly used kernels, as the covariance function $k(\cdot, \cdot)$. With the Radial Basis Function (RBF), we define an $n \times n$ matrix denoted by K where the entry at the i-th row and at the j-th column in K is $k(\mathbf{x}_i, \mathbf{x}_j)$ for $i \in [1, n]$ and $j \in [1, n]$.

Consider the second posterior stage. We define the *posterior mean function* (i.e., $\hat{\eta}(\mathbf{x})$) based on S_f as follows.

Let $\mathbf{f}$ be an n-dimensional vector containing n real numbers which is equal to $\{f_i\}_{i=1}^{n}$ and $\mathbf{I}$ be the $n \times n$ *identity matrix*. We denote $k(\cdot, \cdot)$ to be the *covariance function* which takes two features as inputs and returns a positive real number as an output. We define matrix K of order $n \times n$ where the entry at the i-th row and at the j-th column in K is $k(\mathbf{x}_i, \mathbf{x}_j)$ for $i \in [1, n]$ and $j \in [1, n]$. According to [23], we can express $\hat{\eta}(\mathbf{x})$ as follows.

$$(3.1) \qquad \hat{\eta}(\mathbf{x}) = \alpha^T \mathbf{k}(\mathbf{x}) + 0.5.$$

where α is an n-dimensional vector containing n real numbers, which is equal to $(K + \sigma^2 \mathbf{I})^{-1}(\mathbf{f} - 0.5)$, and $\mathbf{k}(\mathbf{x})$ is an n-dimensional vector containing n covariance functions, which is equal to $\{k(\mathbf{x}, \mathbf{x}_i)\}_{i=1}^{n}$. Let $\alpha = (\alpha_1 \alpha_2 ... \alpha_n)^T$ where α_i is a real number for $i \in [1, n]$. We denote $\|\alpha\|$ to be the L_2*-norm* of α. Given an instance with its feature $\mathbf{x}_i$ where $i = 1, 2, ..., n$, we call $k(\mathbf{x}, \mathbf{x}_i)$ as an *instance-based kernel function*. In the above form of $\hat{\eta}(\mathbf{x})$, we can regard $\hat{\eta}(\mathbf{x})$ as a *weighted linear combination* of n instance-based kernel functions where each value α_i in the vector α is regarded as a *weight*.

Due to the nice property that $\hat{\eta}(\mathbf{x})$ can be expressed as a weighted linear combination, we define the *function class*, denoted by $\mathcal{F}$, to be used later in this paper. It contains all possible functions each of which maps the feature space $\mathbf{x}$ of any instance I to a real number ranging from 0 to 1 and is expressed in a weighted sum of n instance-based kernel functions. That is, for each function $\hat{\eta}(\cdot) \in \mathcal{F}$, it can be written as $\alpha^T \mathbf{k}(\mathbf{x})$, where α is an n-dimensional vector of $\mathbb{R}^n$. In this paper, we consider all functions in the function class where the α vector associated with each function has its L_2-norm value at most a given value A where A is a positive real number given by users. A can be regarded as a parameter describing the complexity of the function class. If A is larger, then the complexity of this class is higher.

Similarly, Var$(\mathbf{x})$ can be expressed as follows [23].

$$(3.2) \text{Var}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T (K + \sigma^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{x})$$

Note that $\hat{\eta}(\mathbf{x})$ is dependent on the fractional scores (i.e., $\mathbf{f}$) but Var$(\mathbf{x})$ is not.

3.2 Strategy In the previous section, we know the formulation of the estimated fractional score, $\hat{\eta}(\mathbf{x})$ (or $\hat{\eta}$ in short), and the variance of this estimated value, Var$(\mathbf{x})$ (or Var in short). In this section, we are ready to address the second issue. That is, we present a strategy to decide whether we should obtain the fractional score of a given instance with its feature $\mathbf{x}$ from an expert.

Note that $\hat{\eta}$ and Var are used to estimate the true η, and their formulations are independent of *sampling* used in the algorithm. Next, we introduce another estimated value $\hat{\eta}_t \in \mathcal{F}$ whose formulation is dependent on sampling for $t \in [1, n]$. $\hat{\eta}_t$ is called the *sample-based estimated value* and is used for sampling.

Before we describe how to find this sample-based estimated value, we define the *probabilistic regularized empirical error*, an error measurement of a given sample-based estimated value $\hat{\eta}_t$. Then, we want to find this sample-based estimated value $\hat{\eta}_t$ which has the smallest probabilistic regularized empirical error.

Probabilistic Regularized Empirical Error: Let t be an integer in the range of [1, n]. Consider that we are determining whether the fractional score of the t-th instance should be obtained. Let $\hat{\eta}_t(\mathbf{x})$ (or $\hat{\eta}_t$ in short) be the *sample-based estimated value (or fractional score)* of an instance *when* we are determining whether the fractional score of the t-th instances are obtained. We denote the sampling probability of the t-th instance by p_t for $t \in [1, n]$. We also denote a variable Q_t which is equal to 1 if the fractional score of the t-th instance is obtained and is equal to 0 otherwise for $t \in [1, n]$.

We define the *probabilistic regularized empirical error* as follows. Given a sample-based estimated value $\hat{\eta}_t$, the *probabilistic regularized empirical error* of $\hat{\eta}_t$, denoted by $J'[\hat{\eta}_t]$, is defined as follows.

$$(3.3) J'[\hat{\eta}_t] = \frac{\sigma^2}{2} \|\hat{\eta}_t\|^2_{\mathcal{H}} + \frac{1}{2} \sum_{i=1}^{t-1} \frac{Q_i}{p_i} (f_i - \hat{\eta}_t(\mathbf{x}_i))^2$$

where $\|\hat{\eta}_t\|^2_{\mathcal{H}} = \sum_{i=1}^{t-1} \sum_{j=1}^{t-1} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$ is the norm in the Reproducing Kernel Hilbert Space (RKHS) [23]. It is easy to verify that the expected value of the functional $J'[\hat{\eta}]$ is exactly the same as that of the original functional $J[\hat{\eta}]$ for any $\hat{\eta}$.

Finding $\hat{\eta}_t$ with Minimum Empirical Error: After we define this empirical error, we obtain the sample-based estimated value $\hat{\eta}_t$ for $t \in [1, n]$ by minimizing the empirical error. That is,

$$(3.4) \qquad \hat{\eta}_t = \arg\min_{\hat{\eta} \in \mathcal{F}} J'[\hat{\eta}]$$

Strategy: Before we introduce the strategy, we define the following notations first. We assume that the true η follows the distribution of $\mathcal{N}(\hat{\eta}, \text{Var})$. Given a real number v, the probability that $\eta = v$ is equal to $\exp(-\frac{(v-\hat{\eta})^2}{2\text{Var}})$.

Given a real number u, the *cumulative distribution function* of the distribution, denoted by $CD(u, \hat{\eta}, \text{Var})$, is equal to $\int_{-\infty}^{u} \exp{(-\frac{(v-\hat{\eta})^2}{2\text{Var}})} dv$.

Specifically, given an instance, after we derive its estimated value $\hat{\eta}$ and its variance Var described in Section 3.1, we will obtain its fractional score from an expert with a *sampling probability* which is calculated based on these two terms together with the sample-based estimated value $\hat{\eta}_t$. The principle of our sampling strategy is based on *uncertainty sampling*. If the sample-based estimated value $\hat{\eta}_t$ (depending on sampling) is very different from the estimated value $\hat{\eta}$ (independent of sampling), its fractional score must be obtained with a high probability. Suppose that they are similar. We consider the following strategy. If $\hat{\eta}$ is close to 0.5, then it is very likely that the true η is near to 0.5 and this instance is very uncertain. In this case, its fractional score will be obtained with a higher probability. If Var is very small, then it is very likely that the true η is equal to $\hat{\eta}$. In this case, its fractional score will be obtained with a lower probability since our estimation is very accurate and there is no need to obtain the fractional score of this instance.

Based on the above strategy, we define the sampling probability, denoted by $\lambda(\hat{\eta}, \text{Var})$, as follows.

$$\lambda(\hat{\eta}, \text{Var}) = \begin{cases} CD(0.5, \hat{\eta}, \text{Var}) & \text{if } \hat{\eta}_t \geq 0.5 \\ 1 - CD(0.5, \hat{\eta}, \text{Var}) & \text{otherwise} \end{cases}$$

In some cases, we want to guarantee that the fractional score of each instance should be obtained with at least a certain probability. We introduce a *smoothing parameter* denoted by p_{min} which is a non-negative real number and corresponds to the minimum probability of obtaining the fractional score of a given instance. When p_{min} is considered, the sampling probability is the maximum value between $\lambda(\hat{\eta}, \text{Var})$ and p_{min}.

3.3 Algorithm Our proposed algorithm for FSAL can be found in Algorithm 1. In this algorithm, we introduce variable S_t for $t \in [1, n]$ denoting the training dataset which contains all instances with fractional scores after we have seen t instances so far. Besides, for initialization, we define three variables, namely S_0, $\hat{\eta}_0(\cdot)$ and $\text{Var}_0(\cdot)$, which are initialized to $\emptyset$, 0.5 and $k(\cdot, \cdot)$, respectively. During the process of the algorithm, S_t, $\hat{\eta}_t(\cdot)$ and $\text{Var}_t(\cdot)$ are updated based on S_{t-1}, $\hat{\eta}_{t-1}(\cdot)$ and $\text{Var}_{t-1}(\cdot)$. Detailed steps can be found in Algorithm 1.

It is easy to verify that our proposed algorithm is tractable because Gaussian Process Regression, the basic component in our algorithm, is tractable, and other operations in our algorithm can also be done in linear time.

Algorithm 1 Algorithm for Fractional-Score-based Active Learning

```
Input: an unlabeled dataset ${\mathbf{x}_1, ..., \mathbf{x}_n}$, a noise parameter $\sigma$, a kernel function
    $k(\cdot, \cdot)$, a smoothing parameter $p_{min}$
Output: a classifier $h$, a training dataset $S_f$
$S_0 \leftarrow \emptyset$; $\hat{\eta}_0(\cdot) \leftarrow 0.5$; $\text{Var}_0(\cdot) \leftarrow k(\cdot, \cdot)$
for $t = 1$ to $n$ do
    calculate the sampling probability, i.e., $p_t = \lambda(\hat{\eta}_{t-1}, \text{Var}_{t-1})$
    flip a coin with two possible outcomes 0 and 1 where P(outcome = 1) = $p_t$
    let $Q_t$ be the outcome of the coin flip
    if $Q_t = 1$ then
        obtain the fractional score $f_t$ of $\mathbf{x}_t$
        $S_t \leftarrow S_{t-1} \cup \{(\mathbf{x}_t, f_t)\}$
        update $\hat{\eta}_t(\cdot)$ according to Equation (3.4); update $\text{Var}_t(\cdot)$ according to
        Equation (3.2)
    else
        $S_t \leftarrow S_{t-1}$; $\hat{\eta}_t \leftarrow \hat{\eta}_{t-1}$; $\text{Var}_t \leftarrow \text{Var}_{t-1}$
    end if
    $h_t(\cdot) \leftarrow \mathbb{I}_{\hat{\eta}_t(\cdot) \geq 0.5}$
end for
return $h_n$ and $S_n$
```

3.4 Theoretical Analysis Concepts and Notations: Based on the function class $\mathcal{F}$, we define the *hypothesis space*, denoted by $\mathcal{H}$, to be $\{\hat{h} : \hat{h} = \mathbb{I}_{\hat{\eta} \geq 0.5}, \hat{\eta} \in \mathcal{F}\}$. Let d be the VC dimension of $\mathcal{H}$. Note that $0 < d < \infty$.

Given a function $\hat{\eta} \in \mathcal{F}$ and an instance-score pair $(\mathbf{x}, f)$ where $\mathbf{x} \in \mathcal{X}$ and $f \in [0, 1]$, we define the *square loss* of $\hat{\eta}$ with respect to this pair, denoted by $\ell_{\hat{\eta}}(\mathbf{x}, f)$, to be $(\hat{\eta}(\mathbf{x}) - f)^2$.

Let X be a random variable denoting the feature of an instance and F be a random variable denoting the fractional score of an instance. Following [2, 4, 13, 18], we assume that all instances are generated according to the joint distribution on two random variables X and F, denoted by $P(X, F)$.

Given $\hat{\eta} \in \mathcal{F}$, we define the *expected loss* of $\hat{\eta}$, denoted by $EL(\hat{\eta})$, to be $\mathbb{E}_{(\mathbf{x},f)\sim P(X,F)}[\ell_{\hat{\eta}}(\mathbf{x}, f)]$. Let $\hat{\eta}^*$ be the optimal function in $\mathcal{F}$. That is, $\hat{\eta}^* = \arg\min_{\hat{\eta} \in \mathcal{F}} EL(\eta)$. Following [17, 23], we assume that $EL(\hat{\eta}^*) = 0$ in order to simplify the proof to be shown later. This assumption is similar to the Bayes classifier assuming that there is no classification error for the classifier. Relaxing this assumption is left as a future work.

In order to give a tight theoretical bound on the error, we capture the *distribution* of all fractional scores in the dataset by the *Margin Assumption* [30] which is commonly adopted in the literature [19, 3, 28, 26]. Intuitively, this assumption states that it is not likely that a fractional score is near to $\frac{1}{2}$. Let $P(X)$ be the distribution on the random variable X that the features of the instances in the dataset follow.

DEFINITION 1. (MARGIN ASSUMPTION) *For any real number ω where $0 < \omega \leq 1$, there exist two constants $\gamma > 0$ and $c > 0$ such that*

$$Pr_{\mathbf{x}\sim P(X)}(|2\eta(\mathbf{x}) - 1| < \omega) \leq c \cdot \omega^{\gamma} \tag{3.5}$$

Both c and γ are the parameters for describing the distribution which the fractional scores in the dataset follow. The margin assumption can be explained as follows. Suppose that c and γ are known. Note that $\eta(\mathbf{x})$ is in the range between 0 and 1. If ω is near to 0, then the inequality in the assumption states that the probability that $\eta(\mathbf{x})$ is close to $\frac{1}{2}$ is very small. If ω is near to 1, then the inequality means that the probability that $\eta(\mathbf{x})$ is close to either 0 or 1 is very large.

As we described before, c and γ are used to describe the distribution of the fractional scores. If c is smaller, then it is less likely that a fractional score is near to $\frac{1}{2}$. If γ is smaller, then ω^{γ} will be larger (when $\omega \in (0, 1)$). In this case, it is more likely that a fractional score is near to $\frac{1}{2}$ (which can be considered that there is more noise in the dataset). Thus, if c is very small and γ is very large, it is less likely that a fractional score is near to $\frac{1}{2}$.

Theoretical Results: In the following, we show our theoretical result.

THEOREM 3.1. (LABEL COMPLEXITY) *Given a confidence parameter $\delta \in (0, 1)$ and an error parameter $\epsilon \in (0, 1)$, if the excess error of the classifier returned by Algorithm 1 is at most ϵ, then with probability at least $1 - \delta$, the expected number of instances whose fractional scores are obtained in Algorithm 1, denoted by T, satisfies the following inequality.*

$$T \leq \frac{\theta}{\epsilon^{\frac{4}{2+\gamma}}} \vee (c \cdot \sum_{t=1}^{n}(1 \wedge (\frac{\theta}{p_{min}t}))^{\gamma/2})$$

where $\theta = 128 \cdot \frac{\ln \mathcal{M}(\epsilon/32, \mathcal{F}) + \ln \frac{2}{\delta}}{p_{min}} + 2\sigma^4 A^2$ *and* $\mathcal{M}(\epsilon/32, \mathcal{F}) = O(\frac{1}{\epsilon})$.

If γ is smaller (i.e., there is more noise in the dataset), then $\epsilon^{-\frac{4}{2+\gamma}}$ is larger (since $\epsilon \in (0, 1)$). Thus, the label complexity is larger. If ϵ is smaller, then it is obvious that the label complexity becomes larger.

As we described before, our proposed algorithm can address the two challenges mentioned in Section 1. That is, our proposed algorithm is tractable. Besides, its label complexity is lower than the label complexity of the best-known tractable active learning algorithm [5].

4 Experiments

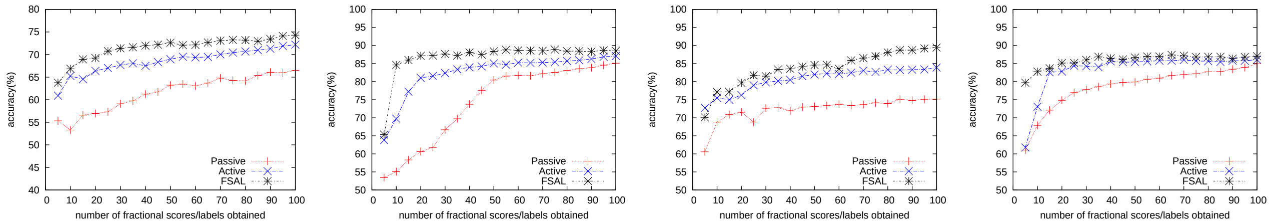
Experimental Setup: We conducted experiments on a workstation with 1.60GHz CPU and 3.06GB RAM. We consider two types of real datasets in our experiments. The first type of real datasets are used for regression originally, while the second type of real datasets are used for classification originally.

The first type of real datasets comes from regression datasets. We used four regression datasets, namely "breast-cancer", "housing", "wine-red" and "wine-white", from the UCI repository [15]. Note that each real dataset is associated with feature attributes and a target attribute in the regression problem. It is obvious that in our problem setting, the feature attributes originally used for regression corresponds to the feature attributes for our problem. The normalized value of the target attribute of each instance originally used for regression, ranging from 0 to 1, corresponds to the probability that the instance belongs to class 1. For example, in dataset "wine-red", the target attribute denotes the quality of the wine, ranging from 1 to 10, in the regression problem. In our problem, the normalized value corresponds to the probability that an instance (wine) has a good quality (which corresponds to class 1). Each dataset containing these feature attributes and the probabilities corresponds to the probabilistic dataset S_o without any noise. This can be regarded as the ground-truth dataset in our problem setting. However, as we described in Section 2, we are only given an "observed" version of the probabilistic dataset, says S_f. Thus, we generate S_f by adding a noise value randomly picked from $\mathcal{N}(0, \sigma^2)$ to each probability. Each added value corresponds to a fractional score in S_f. Note that if the added value is greater than 1, we re-set this value to 1. If it is smaller than 0, we re-set this value to 0. We set the maximum budget for these four datasets be 100.

Besides, we employ a movie review dataset [20] as an example of the second type of real classification datasets. Each movie corresponds to an instance where its feature attributes are the vocabularies from the reviews provided by the IMDb users, its fractional score is the average normalized user rating and its target attribute is the sentiment porality of the movie reviews. This dataset was originally used for Movie Review Sentiment Classification. We picked 30000 movies each of which has over 20 thousands ratings in *IMDb.com* from the original dataset for training, where 7500 movies are labeled as "positive" and the remaining are labeled as "negative". According to the concept of crowdsourcings, the average rating based on a large number of movie fans is rather close to the probability that a person has a positive impression on this movie. Thus, no noise is added to the dataset because we consider that it is an accessible way to gather accurate fractional scores. We set the maximum budget for the dataset be 10000, which means that we sample at most one-third of the movies from the whole dataset.

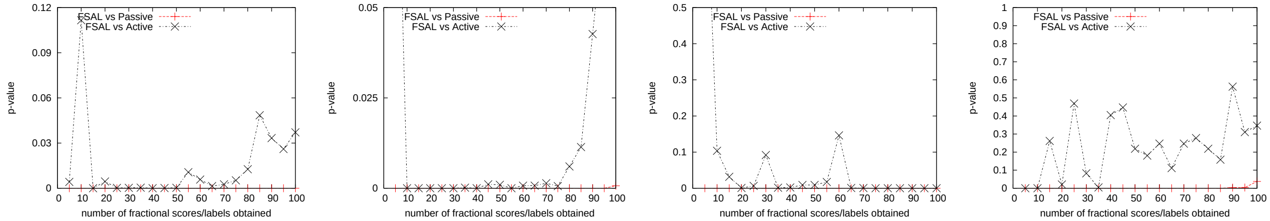
We denote our proposed algorithm in Algorithm 1 based on S_f by *FSAL*. In this algorithm, we adopt the Radial Basis Function (RBF), one of the most commonly used kernels, as the covariance function.

We compared *FSAL* with two other traditional tractable algorithms, namely *Passive* and *Active*. We did not compare the intractable algorithms (e.g., the passive learning [30] with the best-known theoretical bound and the active learning [19] with the best-known theoretical bound) since they are not efficient. Note that since the two traditional tractable algorithms are based on the deterministic dataset, according to S_o, we generate the corresponding deterministic dataset S_c by setting the target attribute value of each instance to 0/1



(a) breast-cancer (b) housing (c) wine-white (d) wine-red

Figure 1: The average accuracy of *Passive*, *Active* and *FSAL*



(a) breast-cancer (b) housing (c) wine-white (d) wine-red

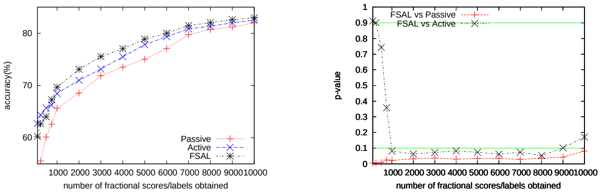
Figure 2: The p-value of two paired t-tests: (*FSAL* vs *Passive*) and (*FSAL* vs *Active*)

labels randomly according to the probabilities in S_o. *Passive* corresponds to a passive learning approach which finds a classifier based on the target attribute values of *all* instances in S_c. We set the sampling probability in our proposed algorithm FSAL to 1 for implementing *Passive*. *Active* corresponds to a traditional passive learning approach which finds a classifier based on the target attribute values of *some* selected instances in S_c. We adopted the importance weighted active learning algorithm [5] for this purpose because *Active* used a similar sampling algorithm as *FSAL* but it is based on S_c instead of S_f.

We set the noise parameter $\sigma = 0.2$ and the smoothing parameter $p_{min} = 0.2$ in the experiments.

We performed a 10-fold *cross-validation* for each algorithm. In particular, the training set S_f was randomly partitioned into 10 pieces, each of which was held out for testing in one of the ten folds, while the remaining nine pieces were collected for training. In the training phase, we did not terminate the algorithm until 100 fractional scores/labels were obtained. In the testing phase, we evaluated the performance of a classifier in terms of its average accuracy on the held-out test set. We repeated the cross-validation four times for each algorithm, so each reported value was an average of 20 results when a certain number of fractional scores/labels were obtained.

Experimental Results: Figure 1 shows our result on different regression training datasets when the number of fractional scores/lables obtained changes. It is obvious that the average accuracy of *FSAL* is much higher than that of *Passive* because *FSAL* chooses some "informative" instances *delicately* but *Passive* choose all instances. Besides, *FSAL* also performs better than *Active* in most circumstances since fractional scores in S_f used by *FSAL* have more informative information about probabilities compared with 0/1 labels in S_c used by *Active*. Figure 2 shows the result of the paired t-test, denoted by "*FSAL* vs A", for the experiments used for Figure 1 to compare the significance of the "difference" between the accuracy of *FSAL* and the accuracy of another traditional algorithm A where the p-value is a measurement in the t-test. When the p-value is close to 0, we say that *FSAL* is better than A in a statistical significance. In the figure, we can find that *FSAL* is much better than *Passive* in all of our experiments, while *FSAL* is better than *Active* in most cases.



(a) The average accuracy (b) The p-value of paired t-test

Figure 3: Performance of different classifiers on the movie review dataset

Furthermore, Figures 3(a) and 3(b) shows our result on the second-type dataset. The same notations mentioned in the previous results are employed in the figures. As the effect of active learning becomes less significant when the size of the training dataset increases, the advantage of *Active* compared with *Passive* becomes less significant.

However, in Figure 3(a), we can see that *FSAL* still performs better than both *Passive* and *Active*. Since as we described before, the fractional scores in the dataset are quite close to the underlying conditional probabilities, it is reasonable that FSAL remains its leading superiority when the size of the training dataset increases. The corresponding p-values are presented in Figure 3(b). When the number of fractional scores obtained increases, the p-values become smaller. When the number of fractional scores obtained is large (e.g., at least 1000), the p-values are smaller than 0.1 in most cases, which means that *FSAL* is much better than *Passive* and *Active* in most cases.

5 Related Work

Selective sampling has been developed for a long time [11], but strict theoretical analysis on its superior performance over passive learning appears until recent years [13, 18, 8]. In general, we can divide the related studies into three categories. They are *Realizable Active Learning*, *Noise-based Active Learning* and *Agnostic Active Learning*. These three kinds of studies analyzed the asymptotic result on either the label complexity or the rate of convergence in active learning according to different assumptions on data generation. Lastly, we describe some practical active learning algorithms.

The strongest result on the improvement of the sample complexity in active learning comes from the realizable setting. In this setting, an intuitive algorithm developed by [11] only selects those instances which enable the hypothesis space to be shrunk after their labels are observed. In the realizable setting, the label complexity for *active* learning which is $O(\ln(\epsilon^{-1}))$ is exponentially faster than that for passive learning which is $O(\epsilon^{-1})$.

Since the realizable setting is impractical in real life, it is interesting to know whether learning based on selective sampling is also strictly better than passive learning when an arbitrary noise is allowed to appear in the training dataset. Agnostic active learning [4], a conservative labeling strategy, shrinks the hypothesis space only after enough labels are observed. In particular, this algorithm removes a hypothesis from the current space after knowing that this hypothesis is suboptimal with high confidence. [18] further gives a strict proof on the sample complexity of the agnostic active learning. [18] showed that the sample complexity critically depends on a quantity, called the *disagreement coefficient*. When this term is bounded, an exponential reduction towards the label complexity can be achieved.

The realizable setting and the agnostic setting discussed above are two extreme cases, where the former is *too* optimistic and the latter is *too* pessimistic. Therefore, some researchers proposed to make an assumption on the noise condition in the process of data generation. It is possible to bridge the gap between these two extreme cases by using a parametric model to describe the noise condition. Tsybakov's margin assumption [30] is one of the most prevalent noise conditions considered in passive learning, assuming that a data point has less chance to be generated around the decision boundary resulting in a large margin which helps to accelerate the learning rate. Based on Tsybakov's margin condition, [19] proved the rate of convergence, which is faster than that in passive learning. Other noise conditions include *bracketing entropy*, *uniform noise*, *benign noise* and so on. In this paper, our work is based on the Tsybakov's margin condition, but our results shows that a even faster rate of convergence is possible based on the probabilistic dataset.

Lastly, there are many practical active learning algorithms [7, 9, 29, 32, 33] in the literature. In [29], the authors designed three different query strategies based on the concept of *margin*, which can be computed via SVMs. However, their work is based on a noise-free setting. In [32] and [33], the authors formulated the active learning problem in terms of transductive experimental design, which can effectively explores the information of unlabeled data. Since we focus on utilizing the information of probabilistic labels, exploring the information of unlabeled data is not the major focus of our paper. In [9], the authors proposed an optimal experimental design approach, which simultaneously considered the discriminant and the geometrical structures in the process of active learning. Similarly, [7] also discovered the discriminant and geometrical structure together in the active learning process, whereas the algorithm in [7] was performed in the manifold adaptive kernel space. Both studies [9, 7] formulated active learning as an optimization problem, whereas they did not show any theoretical guarantee on the rate of convergence in active learning. After modifying the above algorithms properly, we may apply these algorithms to the probabilistic setting studied in our work, which is considered as an interesting future work.

6 Conclusion

In this paper, we consider the scenario that labels are probabilistic, and propose to learn a classifier from probabilistic training dataset, which is more informative than the traditional one. We not only propose an supervised learning algorithm with a selective sampling strategy, which selectively obtains the fractional scores of newly observed instances, but also prove the theoretical bound on the label complexity, which is better than the traditional result. We empirically show that the algorithm outperforms both the traditional passive learning algorithm and the traditional active learning algorithm which learn from the traditional training dataset. In short, our work associates the theoretical aspect of active learning with a practical consideration, promoting active learning to perform efficiently with a theoretical guarantee. In the future, a lot of potential novel works can be studied according to the framework of probabilistic labels. For example, our framework can be further extended to transfer learning, where the prior knowledge on the probabilistic information collected from the other learning tasks conducted before can be partially applied for solving a new learning task quickly. Moreover, in case that the label of each instance is given by multiple labelers, the labelers' expertise model can be considered.

Acknowledgements: The research is supported by grant FSGRF13EG27.

References

[1] E. Agirre and P. Edmonds. *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2006.

[2] M. Anthony and P. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge Univ Pr, 2009.

[3] S. Arlot and P. Massart. Data-driven calibration of penalties for least-squares regression. *The Journal of Machine Learning Research*, 10:245–279, 2009.

[4] M. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 65–72. ACM, 2006.

[5] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 49–56. ACM, 2009.

[6] G. Blanchard, G. Lugosi, N. Vayatis, et al. On the rate of convergence of regularized boosting classifiers. *The Journal of Machine Learning Research*, 4:861–894, 2003.

[7] D. Cai and X. He. Manifold adaptive experimental design for text categorization. *Knowledge and Data Engineering, IEEE Transactions on*, 24(4):707–719, 2012.

[8] R. Castro and R. Nowak. Minimax bounds for active learning. In *Proceedings of the 20th annual conference on learning theory*, pages 5–19. Springer-Verlag, 2007.

[9] C. Chen, Z. Chen, J. Bu, C. Wang, L. Zhang, and C. Zhang. G-optimal design with laplacian regularization. In *AAAI*, 2010.

[10] J. Cid-Sueiro. Proper losses for learning from partial labels. In *Advances in Neural Information Processing Systems 25*, pages 1574–1582, 2012.

[11] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[12] K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. *The Journal of Machine Learning Research*, 9:1757–1774, 2008.

[13] S. Dasgupta. Coarse sample complexity bounds for active learning. *Advances in neural information processing systems*, 18:235, 2006.

[14] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Verlag, 1996.

[15] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[16] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

[17] L. Györfi. *A distribution-free theory of nonparametric regression*. Springer Verlag, 2002.

[18] S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pages 353–360. ACM, 2007.

[19] S. Hanneke. Adaptive rates of convergence in active learning. In *Twenty-Second Annual Conference on Learning Theory*. Citeseer, 2009.

[20] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[21] G. Paab and F. Reichartz. Exploiting Semantic Constraints for Estimating Supersenses with CRFs. In *SDM*, 2009.

[22] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. Estimating labels from label proportions. *The Journal of Machine Learning Research*, 10:2349–2374, 2009.

[23] C. Rasmussen. Gaussian processes in machine learning. *Advanced Lectures on Machine Learning*, pages 63–71, 2004.

[24] S. Rueping. Svm classifier estimation from group probabilities. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 911–918, 2010.

[25] J. Schwartz, R. Johnson, and F. Aepfelbacher. Sensitivity, specificity and accuracy of stress spect myocardial perfusion imaging for detection of coronary artery disease in the distribution of first-order branch vessels, using an anatomical matching of angiographic and perfusion data. *Nuclear medicine communications*, 24(5):543, 2003.

[26] C. Scott and R. Nowak. Minimax-optimal classification with dyadic decision trees. *Information Theory, IEEE Transactions on*, 52(4):1335–1353, 2006.

[27] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring Ground Truth from Subjective Labelling of Venus Images. In *Advances in Neural Information Processing Systems 7*, 1995.

[28] I. Steinwart and C. Scovel. Fast rates for support vector machines. *Learning Theory*, pages 853–888, 2005.

[29] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[30] A. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

[31] F. X. Yu, D. Liu, S. Kumar, T. Jebara, and S.-F. Chang. propotion-svm for learning with label proportions. *arXiv preprint arXiv:1306.0886*, 2013.

[32] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on Machine learning*, pages 1081–1088. ACM, 2006.

[33] K. Yu, S. Zhu, W. Xu, and Y. Gong. Non-greedy active learning for text categorization using convex ansductive experimental design. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 635–642. ACM, 2008.