IEICE Transactions on Information and Systems,
vol.E95-D, no.8, pp.2065–2073, 2012.

Early Stopping Heuristics in Pool-Based Incremental Active Learning for Least-Squares Probabilistic Classifier

Tsubasa Kobayashi
Tokyo Institute of Technology, Japan.

Masashi Sugiyama
Tokyo Institute of Technology, Japan.
sugi@cs.titech.ac.jp
http://sugiyama-www.cs.titech.ac.jp/˜sugi

Abstract

The objective of pool-based incremental active learning is to choose a sample to label from a pool of unlabeled samples in an incremental manner so that the generalization error is minimized. In this scenario, the generalization error often hits a minimum in the middle of the incremental active learning procedure and then it starts to increase. In this paper, we address the problem of early labeling stopping in probabilistic classification for minimizing the generalization error and the labeling cost. Among several possible strategies, we propose to stop labeling when the empirical class-posterior approximation error is maximized. Experiments on benchmark datasets demonstrate the usefulness of the proposed strategy.

Keywords

pool-based incremental active learning, early stopping, least-squares probabilistic classifier, semi-supervised learning.

1 Introduction

Supervised learning is aimed at predicting output values for unknown input points. In a standard setup of supervised learning, pairs of input-output samples are provided for training [36, 10, 1]. On the other hand, in some cases, users are allowed to choose input points at which output values are observed. Such a situation is called *active learning* [25] or *experiment design* [7, 23]. Because of this additional degree of freedom for choosing input-point locations, better generalization performance is expected in the active learning scenario. In particular, when the cost of measuring output values is expensive (in terms of time, human labor, money etc.), active learning is highly useful [21, 38, 35].

Active learning can be categorized into two types depending on the problem setup: the *population-based* setup and the *pool-based* setup. In the population-based setup, users are allowed to locate input points at arbitrary positions in the input domain [39, 16, 27]. On the other hand, in the pool-based setup, users can choose input points only from a pool of input-point candidates [15, 31].

Another categorization of active learning is based on data collecting modes: the *batch/off-line* mode and the *incremental/sequential/online* mode. All input-point locations are chosen at once in batch active learning [17, 33], whereas a single input point (or a small batch of input points) is chosen one by one sequentially in incremental active learning [3, 32].

In this paper, we consider pool-based incremental active learning for classification. In this scenario, it is often observed that the test classification error (a.k.a. the *generalization error*) decreases rapidly in an early stage of incremental active learning; then it hits a minimum at some point and it turns to a gradual increase in the end (see Figure 2). Intuitively, the mechanism of this phenomenon can be explained as follows: In an early stage, "informative" input points are chosen for labeling, which tends to improve the classification performance significantly [20, 34]. However, in the pool-based setup, the number of such informative input points is limited. Thus, after using up all informative input points in the pool, the active learning algorithm starts to choose remaining "less informative" input points, which can cause a slight performance decline in a later stage. Actually, this causes not only a loss of classification performance, but also a waste of labeling costs at the same time—this is critical in active learning scenarios with high sampling costs.

To cope with this problem, we address the problem of "early stopping" for incremental active learning. So far, this issue has been investigated for support vector machines [24, 6] and for natural language processing purposes [41, 18, 2]. In this paper, we focus on a recently proposed probabilistic classifier called the *least-squares probabilistic classifier* (LSPC) [28, 29], which was shown to be computationally much more efficient than kernel logistic regression [10] with comparable accuracy.

For LSPC, we investigate the following three criteria for early stopping:

- The cross-validated classification error.
- The empirical classification error.
- The empirical class-posterior approximation error.

It turns out that the cross-validated classification error is not useful for early stopping purposes because training samples are not independent and identically distributed in the incremental active learning scenario, which causes a strong bias in cross-validation estimates of the classification error [37, 4, 30].

On the other hand, the other two criteria, the empirical classification error and the empirical class-posterior approximation error, turn out to behave oppositely to the true test error. That is, the empirical errors increase in an early stage, they hit maxima at

some point, and then they turn to decrease. This phenomenon can also be explained intuitively in a similar way to the test classification error: informative samples gathered in an early stage are generally difficult to classify, and thus the empirical errors tend to grow. On the other hand, less informative samples gathered in a later stage tend to be classified easily, which results in a decrease in the empirical errors.

Among the two empirical error criteria, we show that the empirical classification error tends to be more unstable. This is because the classification error is measured by the 0/1 loss, which only evaluates the correctness of class predictions, and thus is sensitive to sample choice and noise. On the other hand, the class-posterior approximation error reflects overall approximation quality of class-posterior probabilities, and thus it tends to be more stable and reliable.

Consequently, we propose to terminate incremental active learning when the class-posterior approximation error starts to decrease. Through experiments, we demonstrate that our early stopping strategy improves the classification accuracy by about 10%, only with one third of sampling costs.

The rest of this paper is structured as follows. In Section 2, we formulate the problem of probabilistic classification and review LSPC. In Section 3, we review pool-based incremental active learning strategies for probabilistic classification and numerically illustrate their behavior. We then discuss early stopping methods for incremental active learning in Section 4, and experimentally evaluate their performances in Section 5. Finally, we conclude in Section 6.

2 Probabilistic Classification by LSPC

In this section, we formulate the problem of probabilistic classification and review a recently proposed probabilistic classification method called the *least-squares probabilistic classifier* (LSPC) [28, 29].

2.1 Problem Formulation

Suppose that we are given n training samples consisting of input $\boldsymbol{x}$ and label y:

$$(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}, \quad i = 1, \ldots, n,$$

where $\mathcal{X}$ ($\subset \mathbb{R}^d$) is the input domain, $\mathcal{Y} = \{1, \ldots, c\}$ is the set of class labels, and c denotes the number of classes. We assume that the training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ independently follow a joint probability distribution with probability density $p(\boldsymbol{x}, y)$.

The objective of probabilistic classification is to estimate the class-posterior probability $p(y|\boldsymbol{x})$ from the training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Using the class-posterior probability, we can classify a test sample $\boldsymbol{x}$ into $\widehat{y}$ with confidence $p(\widehat{y}|\boldsymbol{x})$:

$$\widehat{y} := \operatorname*{argmax}_{y \in \mathcal{Y}} p(y|\boldsymbol{x}).$$

2.2 Least-Squares Probabilistic Classifier

For each class $y = 1, \ldots, c$, the class-posterior probability $p(y|\boldsymbol{x})$ is modeled by the following linear model:

$$q(y|\boldsymbol{x};\boldsymbol{\alpha}_y) := \sum_{\ell=1}^{b} \alpha_{y,\ell}\phi_\ell(\boldsymbol{x}) = \boldsymbol{\alpha}_y^\top \boldsymbol{\phi}(\boldsymbol{x}),$$

where b is the number of basis functions, $^\top$ is the transpose,

$$\boldsymbol{\alpha}_y = (\alpha_{y,1}, \ldots, \alpha_{y,b})^\top$$

is a parameter vector, and

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_b(\boldsymbol{x}))^\top$$

is a basis function vector. For example, the *Gaussian kernel* could be used as a basis function:

$$\phi_\ell(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_\ell\|^2}{2\sigma^2}\right) \text{ for } \ell = 1, \ldots, b,$$

where σ is a Gaussian width and $\{\boldsymbol{c}_\ell\}_{\ell=1}^b$ are Gaussian centers randomly chosen from training inputs $\{\boldsymbol{x}_i\}_{i=1}^n$.

Let $p(\boldsymbol{x})$ be the marginal density. Then, the parameter $\boldsymbol{\alpha}_y$ in the model $q(y|\boldsymbol{x};\boldsymbol{\alpha}_y)$ is determined so that the following class-posterior squared error is minimized:

$$\begin{aligned} J_y(\boldsymbol{\alpha}_y) &:= \frac{1}{2}\int (q(y|\boldsymbol{x};\boldsymbol{\alpha}_y) - p(y|\boldsymbol{x}))^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2}\int q(y|\boldsymbol{x};\boldsymbol{\alpha}_y)^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &\quad - p(y)\int q(y|\boldsymbol{x};\boldsymbol{\alpha}_y) p(\boldsymbol{x}|y)\mathrm{d}\boldsymbol{x} + \text{Const.} \\ &= \frac{1}{2}\boldsymbol{\alpha}_y^\top \boldsymbol{H}\boldsymbol{\alpha}_y - \boldsymbol{h}_y^\top \boldsymbol{\alpha}_y + \text{Const.}, \end{aligned}$$

where $\boldsymbol{H}$ and $\boldsymbol{h}_y$ are the $b \times b$ matrix and the b-dimensional vector defined as

$$\begin{aligned} \boldsymbol{H} &:= \int \boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})^\top p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \\ \boldsymbol{h}_y &:= p(y)\int \boldsymbol{\phi}(\boldsymbol{x}) p(\boldsymbol{x}|y)\mathrm{d}\boldsymbol{x}. \end{aligned}$$

Ignoring the constant term, approximating the expectations by corresponding sample averages, and adding a regularizer, we obtain the following training criterion:

$$\widehat{\boldsymbol{\alpha}}_y := \operatorname*{argmin}_{\boldsymbol{\alpha}_y \in \mathbb{R}^b} \left[\frac{1}{2}\boldsymbol{\alpha}_y^\top \widehat{\boldsymbol{H}}\boldsymbol{\alpha}_y - \widehat{\boldsymbol{h}}_y^\top \boldsymbol{\alpha}_y + \frac{\lambda}{2n}\boldsymbol{\alpha}_y^\top \boldsymbol{\alpha}_y\right],$$

where $\lambda \boldsymbol{\alpha}_y^\top \boldsymbol{\alpha}_y/(2n)$ ($\lambda > 0$) is the regularization term, and

$$\widehat{\boldsymbol{H}} := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^\top,$$
$$\widehat{\boldsymbol{h}}_y := \frac{1}{n}\sum_{i:y_i=y} \boldsymbol{\phi}(\boldsymbol{x}_i).$$

The solution $\widehat{\boldsymbol{\alpha}}_y$ can be computed analytically as

$$\widehat{\boldsymbol{\alpha}}_y = \left(\widehat{\boldsymbol{H}} + \frac{\lambda}{n}\boldsymbol{I}_b\right)^{-1} \widehat{\boldsymbol{h}}_y,$$

where $\boldsymbol{I}_b$ denotes the b-dimensional identity matrix.

Rounding up negative outputs to zero and renormalizing the solution, we obtain the final solution $\widehat{p}(y|\boldsymbol{x})$ as follows [40]:

$$\widehat{p}(y|\boldsymbol{x}) = \frac{\max(0, \widehat{\boldsymbol{\alpha}}_y^\top \boldsymbol{\phi}(\boldsymbol{x}))}{\sum_{y'=1}^{c} \max(0, \widehat{\boldsymbol{\alpha}}_{y'}^\top \boldsymbol{\phi}(\boldsymbol{x}))}.$$

This method is called the *least-squares probabilistic classifier* (LSPC). Tuning parameters included in LSPC such as the kernel width σ and the regularization parameter λ can be optimized by cross-validation. LSPC was shown to have comparable classification accuracy to *kernel logistic regression*, while its training is computationally much more efficient [28, 40, 26, 9, 22].

When a new sample $(\boldsymbol{x}_{n+1}, y_{n+1})$ is added to the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the LSPC solution can be updated efficiently, based on the *Sherman-Woodbury-Morrison formula* [8]: for an invertible square matrix $\boldsymbol{A}$ and vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ such that $\boldsymbol{v}^\top \boldsymbol{A}^{-1}\boldsymbol{u} \neq -1$, it holds that

$$\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\top\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^\top\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^\top\boldsymbol{A}^{-1}\boldsymbol{u}}.$$

Let

$$\widehat{\boldsymbol{k}}_n = \widehat{\boldsymbol{P}}_n \boldsymbol{\phi}(\boldsymbol{x}_{n+1}),$$

where

$$\widehat{\boldsymbol{P}}_n := \left(\sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^\top + \lambda \boldsymbol{I}_b\right)^{-1}.$$

Then the LSPC solution $\widehat{\boldsymbol{\alpha}}_y^{(n+1)}$ for $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n+1}$ can be incrementally computed from the LSPC solution $\widehat{\boldsymbol{\alpha}}_y^{(n)}$ for $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ as

$$\widehat{\boldsymbol{\alpha}}_y^{(n+1)} = \widehat{\boldsymbol{\alpha}}_y^{(n)} + \frac{I(y = y_{n+1}) - \boldsymbol{\phi}(\boldsymbol{x}_{n+1})^\top \widehat{\boldsymbol{\alpha}}_y^{(n)}}{1 + \boldsymbol{\phi}(\boldsymbol{x}_{n+1})^\top \widehat{\boldsymbol{k}}_n} \widehat{\boldsymbol{k}}_n,$$

where

$$I(\text{cond}) := \begin{cases} 1 & \text{if 'cond' is true,} \\ 0 & \text{otherwise.} \end{cases}$$

For the next iteration, $\widehat{\boldsymbol{P}}_{n+1}$ is computed as

$$\widehat{\boldsymbol{P}}_{n+1} \leftarrow \widehat{\boldsymbol{P}}_n - \frac{\widehat{\boldsymbol{k}}_n \widehat{\boldsymbol{k}}_n^\top}{1 + \boldsymbol{\phi}(\boldsymbol{x}_{n+1})^\top \widehat{\boldsymbol{k}}_n}.$$

3 Pool-Based Incremental Active Learning for Probabilistic Classification

Let us consider a pool of unlabeled samples

$$\mathcal{U} := \{\boldsymbol{x}'_i\}_{i=1}^{n'},$$

which independently follow $p(\boldsymbol{x})$. The objective of pool-based incremental active learning is to choose a sample $\boldsymbol{x}'$ to label from the pool so that the generalization error is minimized. In this section, we review methods of pool-based incremental active learning for probabilistic classification, and numerically illustrate their behavior for LSPC.

3.1 Incremental Active Learning Strategies for Multi-Class Problems

A simple strategy for incremental active learning when the number of classes is $c = 2$ (i.e., binary classification) is *uncertainty sampling* [19]. That is, a sample $\boldsymbol{x}'$ with $p(y = 1|\boldsymbol{x} = \boldsymbol{x}')$ closest to $1/2$ is chosen as the next input point to label. However, for multi-class problems where $c > 2$, there are various possibilities to define the uncertainty of unlabeled samples. Here, we review two popular strategies: the *entropy-based* strategy [13, 14] and the *best-versus-second-best* strategy [14].

3.1.1 Entropy-Based Strategy

The *entropy* is a measure of uncertainty of a random variable [5]: the entropy of class label y given $\boldsymbol{x}$ is defined as

$$E(\boldsymbol{x}) := -\sum_{y=1}^{c} \widehat{p}(y|\boldsymbol{x}) \log \widehat{p}(y|\boldsymbol{x}),$$

where we regard $p \log p$ as 0 if $p = 0$. A larger entropy value implies more uncertainty.

The entropy-based strategy for incremental active learning chooses the unlabeled sample that has the largest entropy [13, 14]:

$$\boldsymbol{x}' := \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{U}} E(\boldsymbol{x}).$$

3.1.2 Best-Versus-Second-Best Strategy

A weakness of the entropy-based strategy is that its choice is heavily influenced by less important classes. To illustrate this, let us consider class-posterior probabilities of two samples $\boldsymbol{x}'_1$ and $\boldsymbol{x}'_2$ for a 10-class classification problem. In the example plotted in Figure 1, $\boldsymbol{x}'_1$ has a significantly higher class-posterior probability for class 3 than other classes, and thus $\boldsymbol{x}'_1$ is less uncertain. On the other hand, the highest class-posterior probability of $\boldsymbol{x}'_2$ (i.e., class 8) is close to the second highest one (i.e., class 7), and thus $\boldsymbol{x}'_2$ is more uncertain. However, because $\boldsymbol{x}'_1$ has a larger entropy than $\boldsymbol{x}'_2$, $\boldsymbol{x}'_1$ is chosen as the next sample to label by the entropy-based strategy. Such a phenomenon tends to be more significant when the number of classes is large.

To cope with this problem, the *best-versus-second-best* (BvSB) strategy was proposed [14]. BvSB chooses the unlabeled sample that has the minimal difference between the highest and the second highest class-posterior probabilities:

$$\boldsymbol{x}' := \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{U}} D(\boldsymbol{x}),$$

where

$$\begin{aligned} D(\boldsymbol{x}) &:= \widehat{p}(y'|\boldsymbol{x}) - \widehat{p}(y''|\boldsymbol{x}), \\ y' &:= \operatorname*{argmax}_{y \in \mathcal{U}} \widehat{p}(y|\boldsymbol{x}), \\ y'' &:= \operatorname*{argmax}_{y \in \mathcal{U} \setminus y'} \widehat{p}(y|\boldsymbol{x}). \end{aligned}$$

In the example illustrated in Figure 1, $\boldsymbol{x}'_2$ is chosen by the BvSB strategy, which would be appropriate.

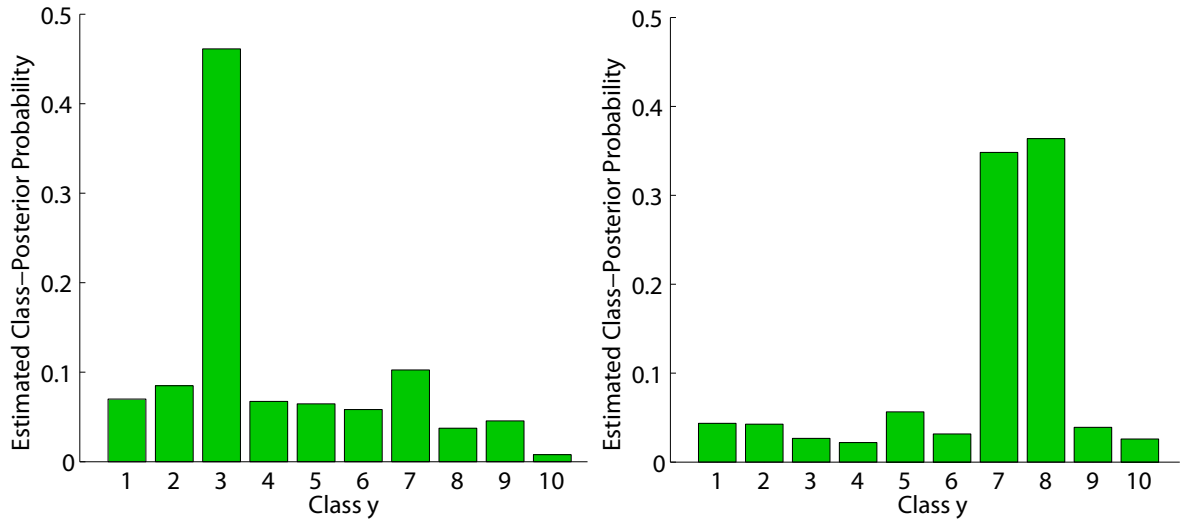
3.2 Numerical Examples

Here, we illustrate how the above incremental active learning strategies behave for LSPC.

3.2.1 Setting

We compare the following three sampling methods:

- **Passive:** Choose an unlabeled sample from the pool just randomly.
- **Active(EP):** Choose an unlabeled sample from the pool using the entropy-based strategy.
- **Active(BvSB):** Choose an unlabeled sample from the pool using the BvSB strategy.



(a) $p(y|\boldsymbol{x} = \boldsymbol{x}'_1)$ (b) $p(y|\boldsymbol{x} = \boldsymbol{x}'_2)$

Figure 1: Class-posterior probabilities of two unlabeled samples $\boldsymbol{x}'_1$ and $\boldsymbol{x}'_2$ for a 10-class problem. $E(\boldsymbol{x}'_1) = 1.81$, $D(\boldsymbol{x}'_1) = 0.359$, $E(\boldsymbol{x}'_2) = 1.68$, and $D(\boldsymbol{x}'_2) = 0.015$.

We employ LSPC with Gaussian kernels as a classifier. The Gaussian width σ and the regularization parameter λ are chosen based on 2-fold cross-validation from

$$\sigma \in \{\tfrac{1}{10}m, \tfrac{1}{5}m, \tfrac{1}{2}m, \tfrac{2}{3}m, m, \tfrac{3}{2}m, 2m, 5m, 10m\},$$
$$\lambda \in \{10^{-3}, 10^{-2.5}, 10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^{0}\},$$

where

$$m := \mathrm{median}(\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\}_{i,j=1}^{n}).$$

We evaluate the classification accuracy of each sampling method using the following 6 classification benchmark datasets (c denotes the number of classes and d denotes the dimensionality of $\boldsymbol{x}$):

- *Satimage* ($c = 6$ and $d = 36$).
- *Pendigits* ($c = 10$ and $d = 16$).
- *Optdigits* ($c = 10$ and $d = 64$).
- *Usps* ($c = 10$ and $d = 256$).
- *Mnist* ($c = 10$ and $d = 717$).
- *Letter* ($c = 26$ and $d = 16$).

Input samples were normalized in the element-wise manner so that each element has mean zero and unit variance.

Initially, a classifier was trained by the LSPC algorithm using training samples consisting of 2 randomly chosen labeled samples from each class. Then, each sampling strategy is applied and the next unlabeled sample to label is chosen from a pool consisting of 100 randomly chosen unlabeled samples from each class. In each iteration, the classification accuracy is evaluated using 100 randomly chosen test samples from each class.

3.2.2 Results

Figure 2 shows the mean misclassification rate over 100 trials for each sampling method. Overall, Active(EP) is comparable to or slightly better than Passive, whereas Active(BvSB) clearly outperforms Passive and Active(EP) for all datasets. In particular, in an early stage of incremental active learning iterations, Active(BvSB) provides a remarkable decrease in the misclassification rate.

However, in a later stage, the misclassification rate of Active(BvSB) tends to increase; for all datasets, we can observe minima of the misclassification rate before all unlabeled samples in the pool are labeled. This phenomenon can be interpreted as follows: Most uncertain unlabeled samples are chosen in the beginning of the incremental active learning process. By this, estimates of the class-posterior probability around decision boundaries are fine-tuned and thus the classification accuracy is improved considerably. However, in a later stage, the algorithm starts to select less uncertain unlabeled samples because no uncertain sample remains in the pool. Such less uncertain samples may disturb the classification performance.

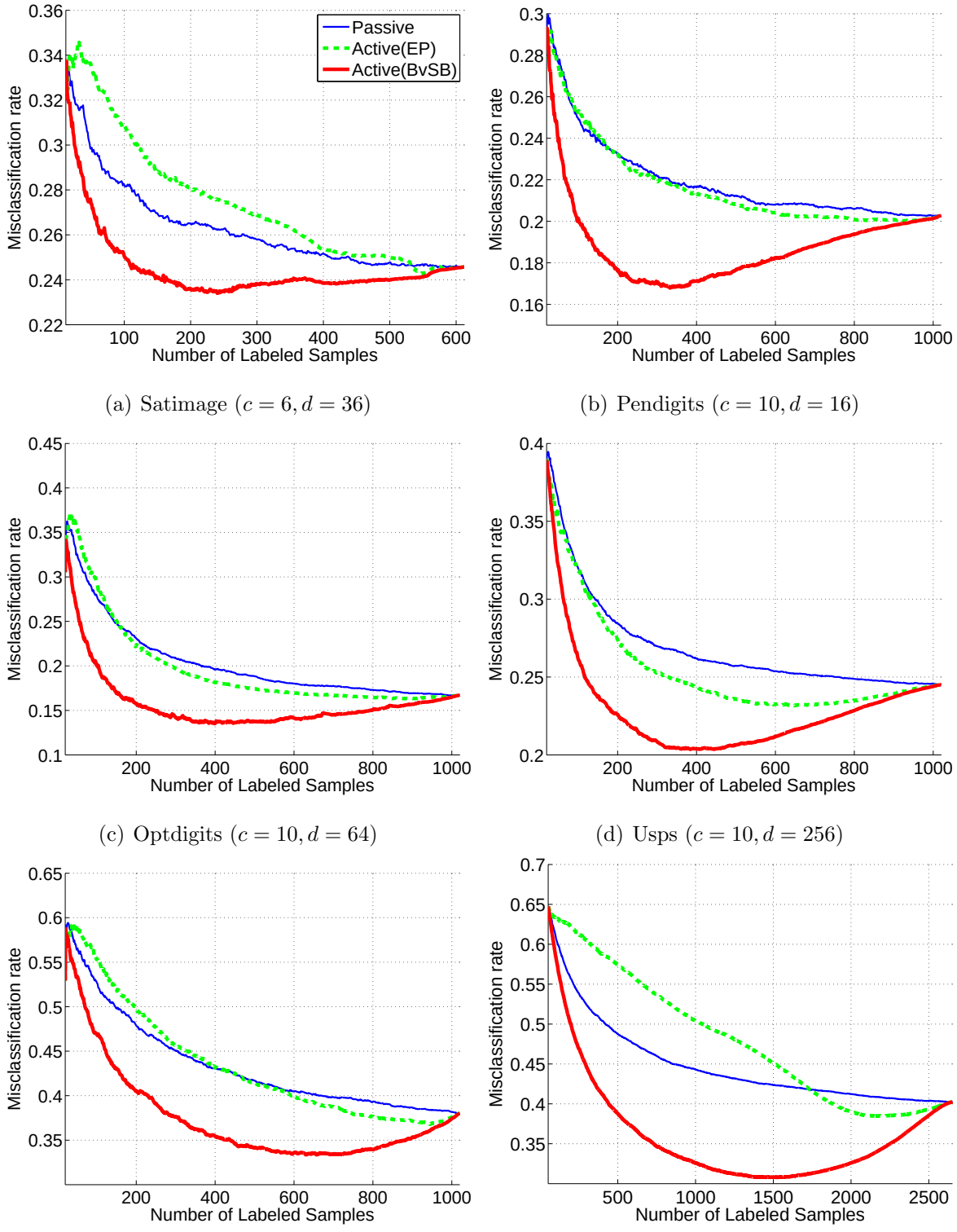
This fact motivates us to *stop* the incremental active learning process halfway, which can reduce the misclassification as well as the sampling cost. In the next section, we investigate how we can appropriately terminate the incremental active learning process.

4 Early Stopping of Incremental Active Learning

As we have shown in the previous section, incremental active learning with the BvSB strategy leads to a remarkable decrease of the misclassification rate in an early stage. However, the misclassification rate tends to increase in a later stage of the incremental active learning process. In this section, we address the problem of early stopping of incremental active learning.

4.1 Cross-Validation Score

A popular method for predicting the misclassification rate is *cross-validation* (CV) [37, 4, 30]. Let us split the labeled training set $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into k disjoint subsets $\{\mathcal{L}_j\}_{j=1}^k$.



(a) Satimage ($c = 6, d = 36$)

(b) Pendigits ($c = 10, d = 16$)

(c) Optdigits ($c = 10, d = 64$)

(d) Usps ($c = 10, d = 256$)

(e) Mnist ($c = 10, d = 717$)

(f) Letter ($c = 26, d = 16$)

Figure 2: Mean misclassification rate over 100 trials for each sampling method.

Then the misclassification rate is predicted by CV as

$$\frac{1}{k}\sum_{j=1}^{k}\frac{1}{|\mathcal{L}_j|}\sum_{(\boldsymbol{x}',y')\in\mathcal{L}_j} I(y' \neq \operatorname*{argmax}_y \widehat{p}_j(y|\boldsymbol{x}')),$$

where $\widehat{p}_j(y|\boldsymbol{x})$ is a class-posterior probability estimated from $\mathcal{L}\backslash\mathcal{L}_j$ and $|\mathcal{L}_j|$ denotes the number of elements in the subset $\mathcal{L}_j$.

Figure 3 shows the mean misclassification rate predicted by 2-fold CV over 100 trials in passive learning and active learning (with the BvSB strategy). In passive learning, the shape of the true test classification error is rather accurately estimated by the CV score for all datasets. However, in active learning, CV performs poorly—the predicted misclassification rate sharply increases in the beginning, and then it monotonically decreases. This phenomenon is induced by selecting samples near decision boundaries in the beginning; because such samples are difficult to correctly classify, and the CV tends to overestimate the true misclassification error. After selecting up all unlabeled samples near decision boundaries, ones far from the boundaries (which can be easily classified) start to be selected and therefore the CV score rapidly decreases.

Theoretically, CV gives an almost unbiased estimate of the misclassification rate, given that training samples are independent and identically distributed [37]. However, in incremental active learning scenarios, training samples are chosen one by one in an sequential manner and thus they are no longer independent of each other. This is the reason why CV is not reliable in incremental active learning scenarios.

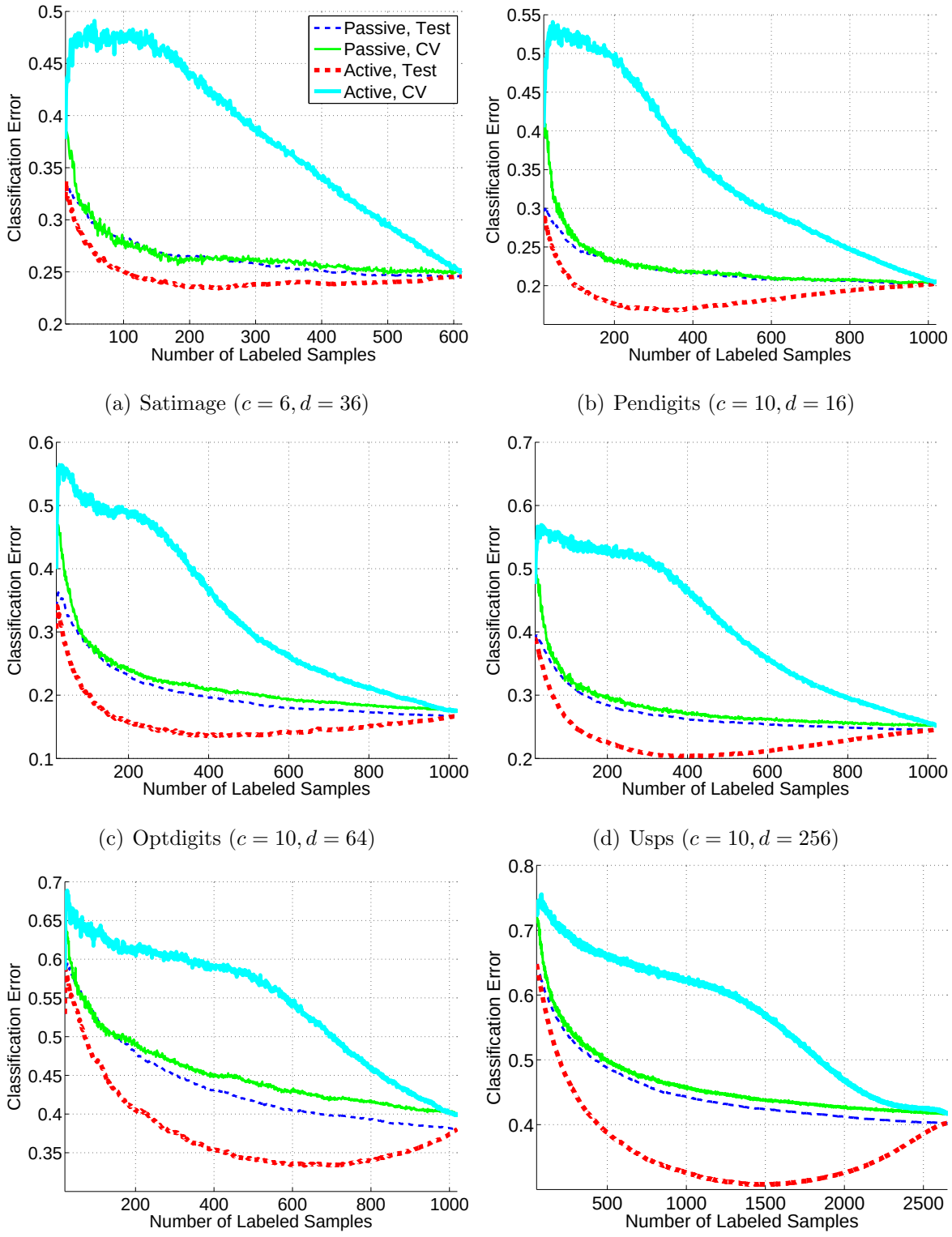
4.2 Empirical Classification Error

As an alternative approach, let us consider the empirical classification error [12], which can be computed using labeled training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ as

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \operatorname*{argmax}_y \widehat{p}(y|\boldsymbol{x}_i)).$$

Figure 4 shows the mean test classification error and mean empirical classification error over 100 trials for passive learning and active learning (with the BvSB strategy). In passive learning, the empirical classification error tends to increase as the number of labeled samples increases, whereas the test classification error tends to decrease.

On the other hand, in active learning, the empirical classification error tends to increase more sharply than passive learning, and then it starts to decrease. This happens because, in the case of active learning, uncertain samples tend to be labeled in the beginning, which are difficult to classify. Thus, the empirical classification error tends to grow significantly. However, in a later stage of the incremental active learning procedure, all uncertain (i.e., difficult to classify) samples have already been labeled, and thus easy-to-classify samples are chosen. For this reason, the empirical classification error tends to decrease.



(a) Satimage ($c = 6, d = 36$)

(b) Pendigits ($c = 10, d = 16$)

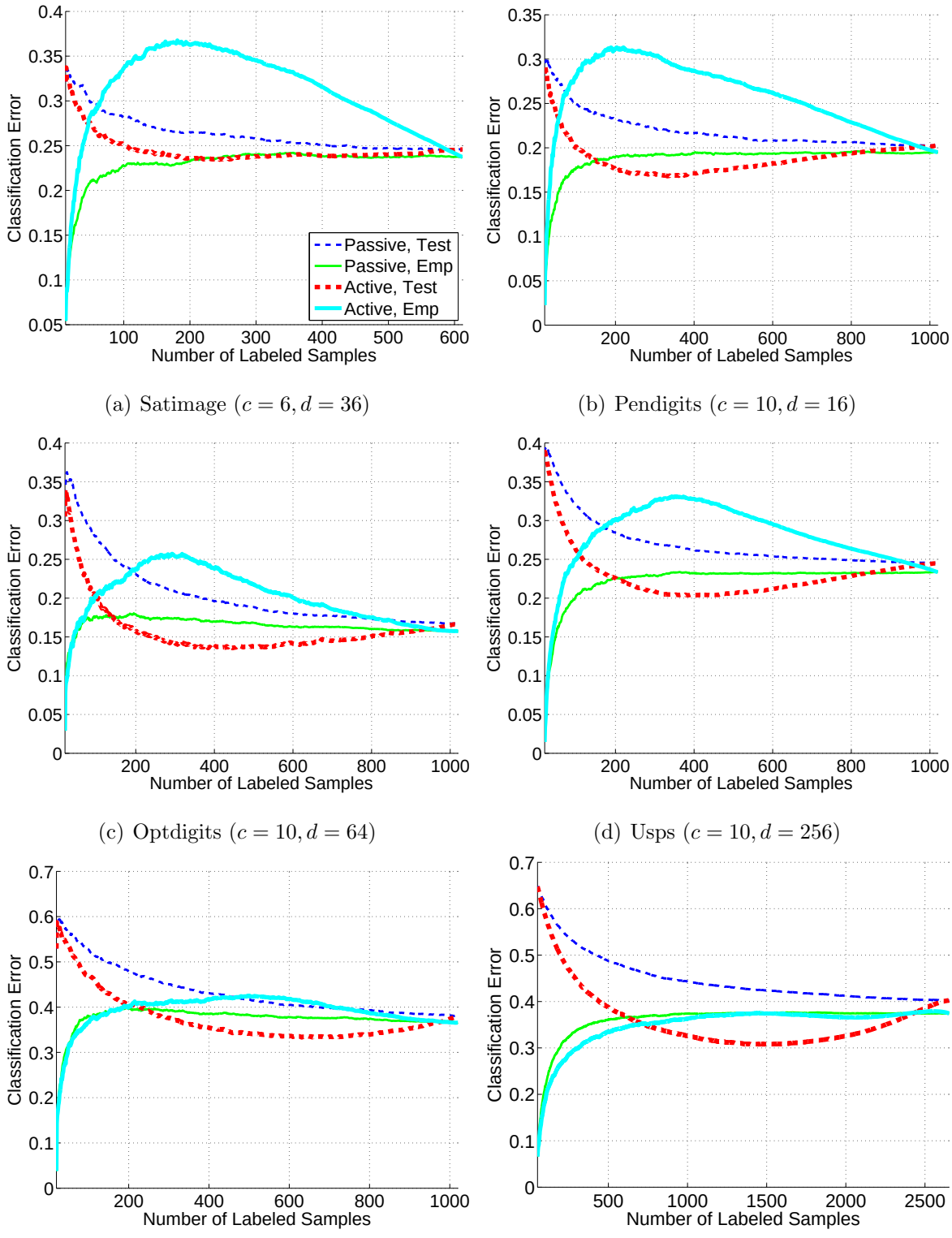
(c) Optdigits ($c = 10, d = 64$)

(d) Usps ($c = 10, d = 256$)

(e) Mnist ($c = 10, d = 717$)

(f) Letter ($c = 26, d = 16$)

Figure 3: Mean classification errors measured on test samples and predicted by 2-fold cross-validation over 100 trials.



(a) Satimage ($c = 6, d = 36$)

(b) Pendigits ($c = 10, d = 16$)

(c) Optdigits ($c = 10, d = 64$)

(d) Usps ($c = 10, d = 256$)

(e) Mnist ($c = 10, d = 717$)

(f) Letter ($c = 26, d = 16$)

Figure 4: Mean test classification error and mean empirical classification error over 100 trials.

The graphs also show that t he behavior of the empirical classification errors in active learning is roughly symmetric to those of the true test classification errors. More importantly, we can see that peaks of the empirical classification error are located around the minima of the true test classification errors. This observation implies that terminating incremental active learning when the empirical classification error starts to decrease is a sensible strategy.

4.3 Empirical Class-Posterior Squared Error

Recall that LSPC learns the class-posterior probability so that the squared error to the true class-posterior probability is minimized. As explained in Section 2.2, the empirical class-posterior squared error (without an irrelevant constant) is given as

$$\sum_{y=1}^{c}\left(\frac{1}{2}\boldsymbol{\alpha}_y^\top\widehat{\boldsymbol{H}}\boldsymbol{\alpha}_y-\widehat{\boldsymbol{h}}_y^\top\boldsymbol{\alpha}_y\right).$$

Figure 5 shows the mean test class-posterior squared error and mean empirical class-posterior squared error over 100 trials for passive learning and active learning (with the BvSB strategy). Note that values can be negative because an irrelevant positive constant is ignored (see Section 2.2). The graphs show that, similarly to the empirical classification error, the empirical class-posterior squared error in active learning also sharply increases compared with passive learning, and then it starts to decrease gradually.

Figure 6 and Figure 7 show the empirical classification error and the empirical class-posterior squared error for a *single trial*. The graph shows that the empirical classification error is more fluctuated than the empirical class-posterior squared error. This is because the classification error is measured by the 0/1 loss, which only evaluates the correctness of class predictions. On the other hand, the class-posterior squared error reflects overall approximation quality of class-posterior probabilities, and thus it tends to be more stable than the 0/1 classification error.

Based on the above discussions, we propose to use the empirical class-posterior squared error as a stopping criterion for active learning.

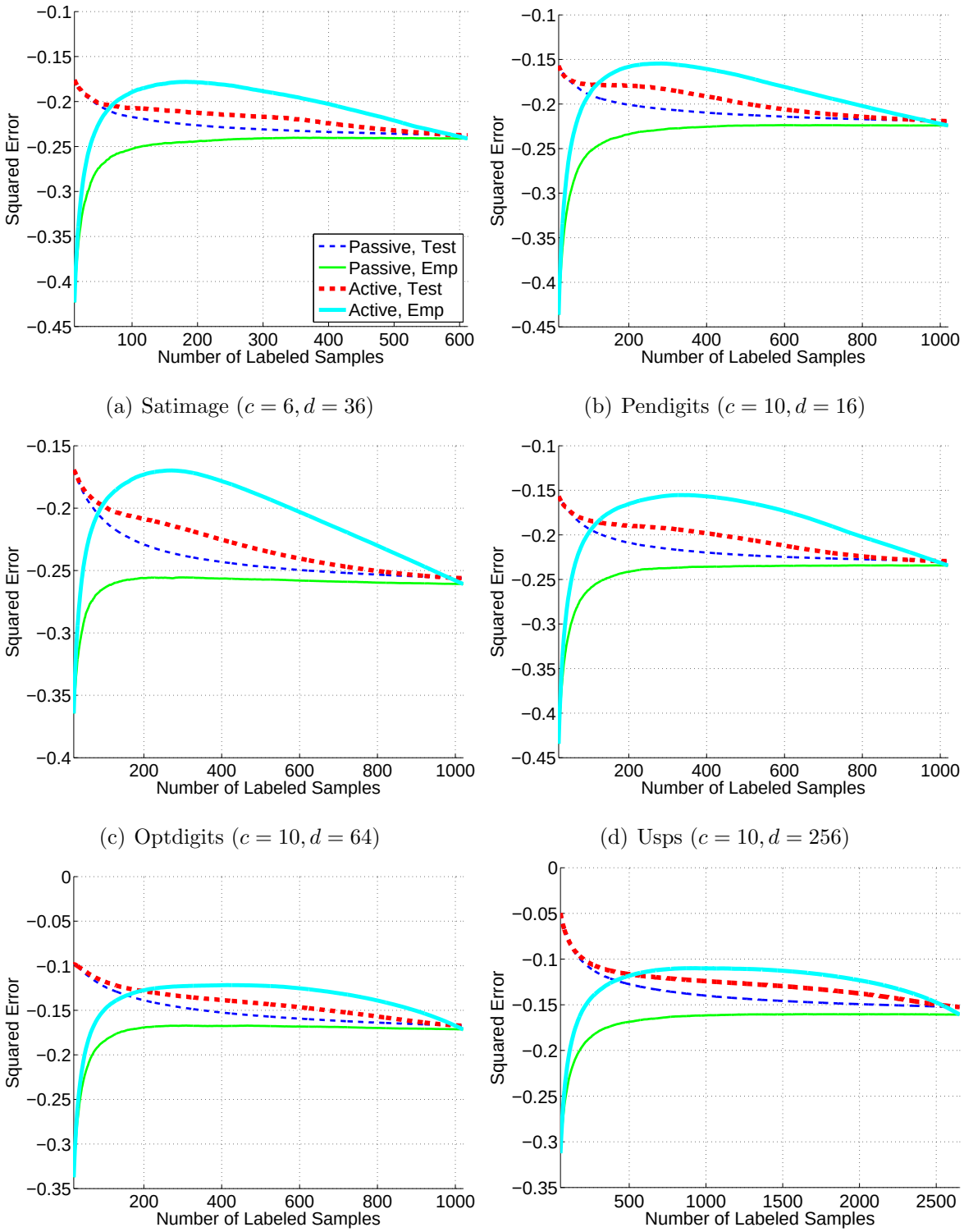
5 Experiments

In this section, we experimentally evaluate the performance of early stopping criteria for incremental active learning.

We use BvSB as the active learning strategy (see Section 3.1.2), and other experimental settings such as datasets are common to Section 3.2.

We evaluate the classification accuracy and the number of labeled samples for the following three methods:

- **ECE:** Terminate active learning if the empirical classification error drops from the value $10c$ samples before.



(a) Satimage ($c = 6, d = 36$)

(b) Pendigits ($c = 10, d = 16$)

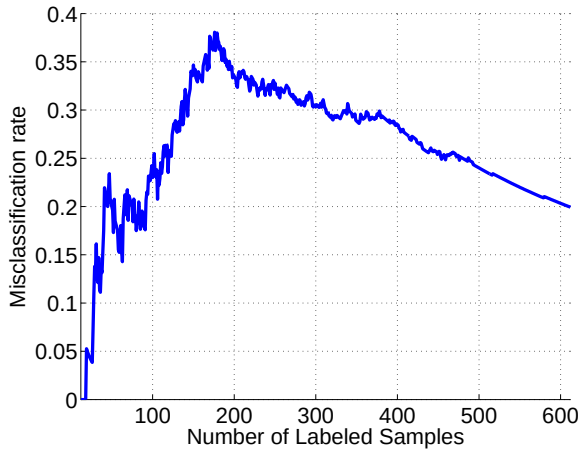
(c) Optdigits ($c = 10, d = 64$)

(d) Usps ($c = 10, d = 256$)

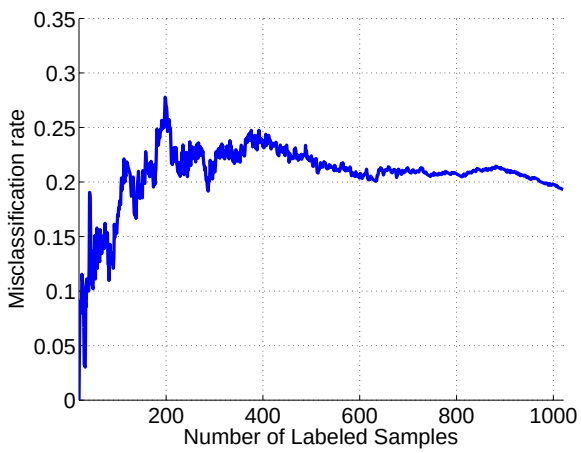
(e) Mnist ($c = 10, d = 717$)

(f) Letter ($c = 26, d = 16$)

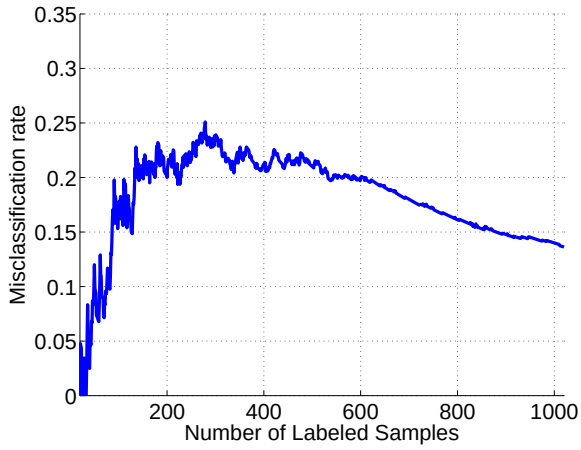
Figure 5: Mean test class-posterior squared error and mean empirical class-posterior squared error over 100 trials. Values can be negative because an irrelevant positive constant is ignored.



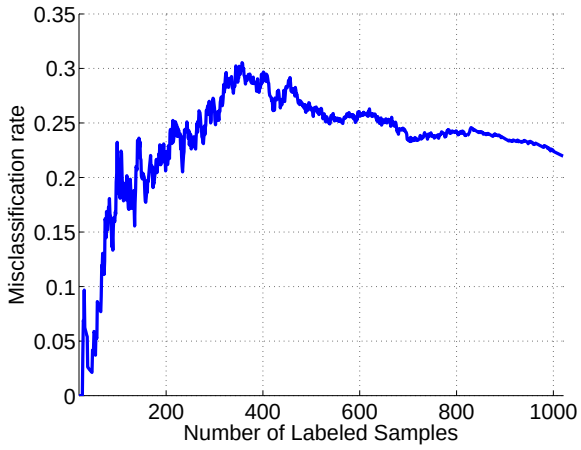
(a) Satimage ($c = 6, d = 36$)



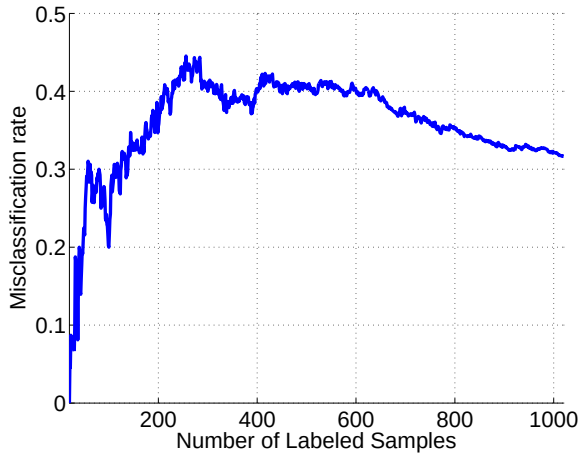
(b) Pendigits ($c = 10, d = 16$)



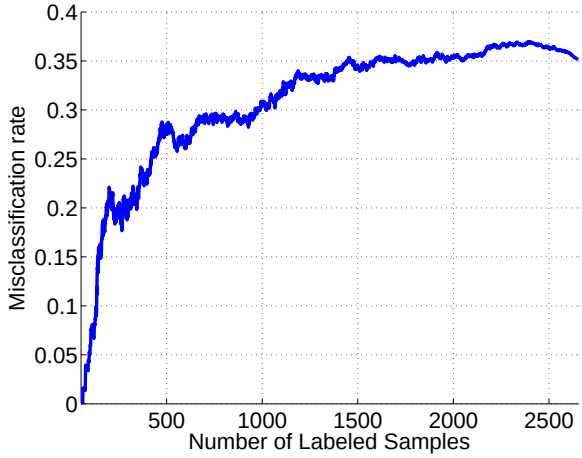
(c) Optdigits ($c = 10, d = 64$)



(d) Usps ($c = 10, d = 256$)

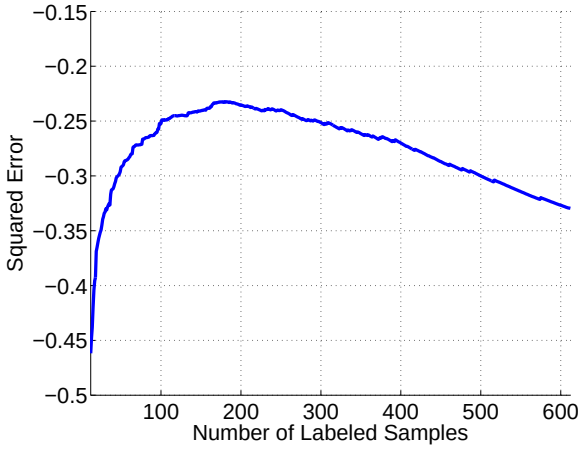


(e) Mnist ($c = 10, d = 717$)

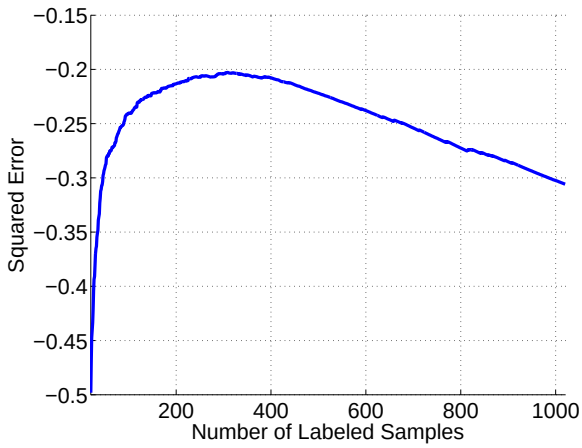


(f) Letter ($c = 26, d = 16$)

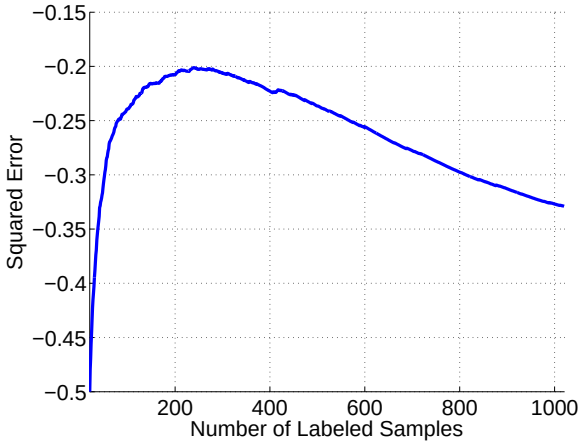
Figure 6: Empirical classification error for a single trial.



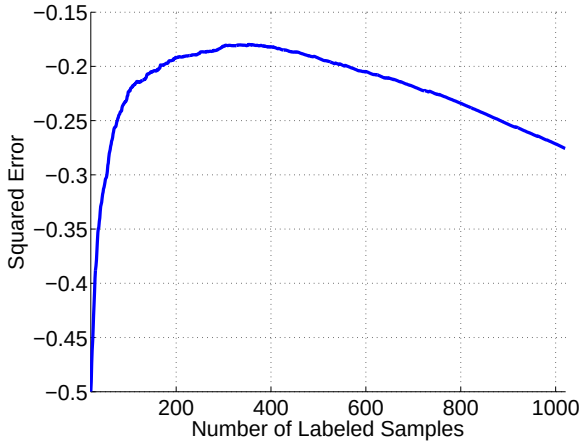
(a) Satimage ($c = 6, d = 36$)



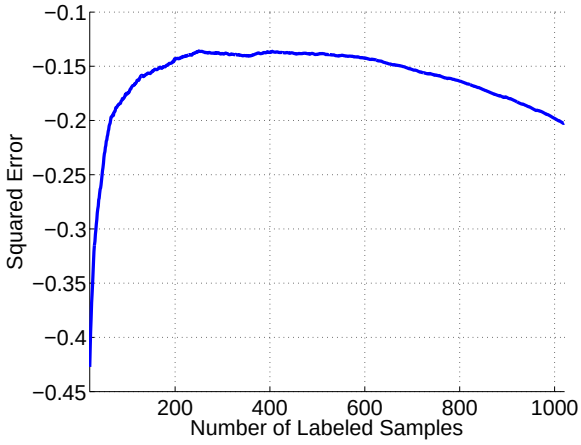
(b) Pendigits ($c = 10, d = 16$)



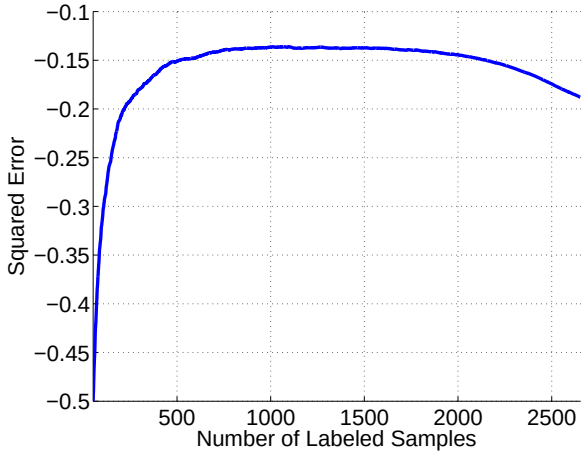
(c) Optdigits ($c = 10, d = 64$)



(d) Usps ($c = 10, d = 256$)



(e) Mnist ($c = 10, d = 717$)



(f) Letter ($c = 26, d = 16$)

Figure 7: Empirical class-posterior squared error for a single trial.

Table 1: Mean misclassification rate over 100 runs.

Method	ECE	ESE	ALL
Satimage	0.2493	**0.2447**	**0.2457**
Pendigits	0.1833	**0.1781**	0.2027
Optdigits	0.1707	**0.1597**	**0.1672**
Usps	0.2243	**0.2093**	0.2450
Mnist	0.4179	**0.3825**	**0.3804**
Letter	0.3662	**0.3277**	0.4024
Average	0.2686	**0.2503**	0.2739

Table 2: Mean percentage of selected samples over 100 runs.

Method	ECE	ESE	ALL
Satimage	20.14	31.68	100.00
Pendigits	18.04	31.61	100.00
Optdigits	17.91	31.68	100.00
Usps	20.85	38.49	100.00
Mnist	18.30	38.71	100.00
Letter	40.01	40.01	100.00
Average	19.88	35.36	100.00

- **ESE:** Terminate active learning if the empirical class-posterior squared error drops from the value $10c$ samples before.
- **ALL:** Label all samples in the pool (no termination).

Note that we introduced the "$10c$"-buffer to avoid terminating active learning just by a small fluctuation of a stopping criterion.

The experimental results are summarized in Table 1 and Table 2, showing the mean misclassification rate and mean percentage of selected samples over 100 trials for each dataset. The best method in terms of the mean misclassification rate and comparable methods according to the *t-test* at the significance level 5% [11] are specified by bold face.

ECE tends to stop the active learning process earlier than ESE (see Table 2), but this seems to be too early because the misclassification error is not reduced enough (see Table 1). This is caused by high fluctuation of the classification error (see Figure 6). On the other hand, the ESE method tends to give smaller misclassification errors than ALL, with only 30–40% labeling costs.

6 Conclusions

In this paper, we proposed an early stopping heuristics for pool-based incremental active learning in least-squares probabilistic classification. Our idea was to terminate the

incremental active learning procedure when the empirical class-posterior approximation error starts to decrease. An intuition behind this heuristic is that if the empirical class-posterior approximation error starts to decrease, all informative samples in the pool have already been used up. Thus, it is a waste of sampling costs to further label unlabeled data. Through experiments, we demonstrated that the proposed heuristic improves the classification accuracy by about 10%, only with one third of sampling costs.

In experiments, we introduced a buffer to avoid terminating active learning just by a small fluctuation of a stopping criterion. How to optimally control the buffer size is left as a future work. Finally, the most important future challenge is to justify our early stopping heuristics theoretically.

Acknowledgments

MS was supported by MEXT KAKENHI 23120004.

References

[1] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, NY, USA, 2006.

[2] M. Bloodgood and K. Vijay-Shanker, "A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping," Proceedings of the Thirteenth Conference on Computational Natural Language Learning, pp.39–47, 2009.

[3] G.E.P. Box and W.G. Hunter, "Sequential design of experiments for nonlinear models," Proceedings of IBM Scientific Computing Symposium in Statistics, pp.113–137, 1965.

[4] D.A. Cohn, "Minimizing statistical bias with queries," Advances in Neural Information Processing Systems 9, ed. M.C. Mozer, M.I. Jordan, and T. Petsche, pp.417–423, The MIT Press, 1997.

[5] T.M. Cover and J.A. Thomas, Elements of Information Theory, 2nd ed., John Wiley & Sons, Inc., Hoboken, NJ, USA, 2006.

[6] S. Ertekin, J. Huang, L. Buttou, and C.L. Giles, "Learning on the border: Active learning in imbalanced data classification," Proceedings of 16th ACM Conference on Information and Knowledge Management, pp.127–136, 2007.

[7] V.V. Fedorov, Theory of Optimal Experiments, Academic Press, New York, NY, USA, 1972.

[8] G.H. Golub and C.F.V. Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[9] H. Hachiya, M. Sugiyama, and N. Ueda, "Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition," Neurocomputing, vol.80, pp.93–101, 2012.

[10] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, NY, USA, 2001.

[11] R.E. Henkel, Tests of Significance, SAGE Publication, Beverly Hills, CA, USA., 1976.

[12] S.S. Ho and H. Wechsler, "Transductive confidence machine for active learning," Proceedings of the International Joint Conference on Neural Networks, pp.1435–1440, 2003.

[13] F. Jing, M. Li, H.J. Zhang, and B. Zhang, "Entropy-based active learning with support vector machines for content-based image retrieval," Proceedings of IEEE International Conference on Multimedia and Expo (ICME2004), pp.85–88, 2004.

[14] A.J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2009), pp.2372–2379, 2009.

[15] T. Kanamori, "Pool-based active learning with optimal sampling distribution and its information geometrical interpretation," Neurocomputing, vol.71, no.1–3, pp.353–362, 2007.

[16] T. Kanamori and H. Shimodaira, "Active learning algorithm using the maximum weighted log-likelihood estimator," Journal of Statistical Planning and Inference, vol.116, no.1, pp.149–162, 2003.

[17] J. Kiefer, "Optimum experimental designs," Journal of the Royal Statistical Society, Series B, vol.21, pp.272–304, 1959.

[18] F. Laws and H. Schütze, "Stopping criteria for active learning of named entity recognition," Proceedings of the 22nd International Conference on Computational Linguistics, pp.465–472, 2008.

[19] M. Li and I. Sethi, "Confidence-based active learning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.28, pp.1251–1261, 2006.

[20] D.J.C. MacKay, "Information-based objective functions for active data selection," Neural Computation, vol.4, no.4, pp.590–604, 1992.

[21] A. McCallum and K. Nigam, "Employing EM in pool-based active learning for text classification," Proceedings of the 15th International Conference on Machine Learning, 1998.

[22] H. Nam, H. Hachiya, and M. Sugiyama, "Computationally efficient multi-label classification by least-squares probabilistic classifier," Proceedings of 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2012), Kyoto, Japan, pp.2077–2080, Mar. 25–30 2012.

[23] F. Pukelsheim, Optimal Design of Experiments, Wiley, New York, NY, USA, 1993.

[24] G. Schon and D. Cohn, "Less is more: Active learning with support vector machines," Proceedings of the Seventeenth International Conference on Machine Learning, pp.839–846, 2000.

[25] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[26] J. Simm, M. Sugiyama, and T. Kato, "Computationally efficient multi-task learning with least-squares probabilistic classifiers," IPSJ Transactions on Computer Vision and Applications, vol.3, pp.1–8, 2011.

[27] M. Sugiyama, "Active learning in approximately linear regression based on conditional expectation of generalization error," Journal of Machine Learning Research, vol.7, pp.141–166, Jan. 2006.

[28] M. Sugiyama, "Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting," IEICE Transactions on Information and Systems, vol.E93-D, no.10, pp.2690–2701, 2010.

[29] M. Sugiyama, H. Hachiya, M. Yamada, J. Simm, and H. Nam, "Least-squares probabilistic classifier: A computationally efficient alternative to kernel logistic regression," Proceedings of International Workshop on Statistical Machine Learning for Speech Processing (IWSML2012), Kyoto, Japan, pp.1–10, Mar. 31 2012.

[30] M. Sugiyama, M. Krauledat, and K.R. Müller, "Covariate shift adaptation by importance weighted cross validation," Journal of Machine Learning Research, vol.8, pp.985–1005, May 2007.

[31] M. Sugiyama and S. Nakajima, "Pool-based active learning in approximate linear regression," Machine Learning, vol.75, no.3, pp.249–274, 2009.

[32] M. Sugiyama and H. Ogawa, "Incremental active learning for optimal generalization," Neural Computation, vol.12, no.12, pp.2909–2940, 2000.

[33] M. Sugiyama and H. Ogawa, "Active learning for optimal generalization in trigonometric polynomial models," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E84-A, no.9, pp.2319–2329, 2001.

[34] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," Journal of Machine Learning Research, vol.2, pp.45–66, March 2002.

[35] K. Ueki, M. Sugiyama, and Y. Ihara, "A semi-supervised approach to perceived age prediction from face images," IEICE Transactions on Information and Systems, vol.E93-D, no.10, pp.2875–2878, 2010.

[36] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, NY, USA, 1998.

[37] G. Wahba, Spline Models for Observational Data, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1990.

[38] M.K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen, "Active learning with SVMs in the drug discovery process," Chemical Information and Computer Sciences, vol.43, no.2, pp.667–673, 2003.

[39] D.P. Wiens, "Robust weights and designs for biased regression models: Least squares and generalized M-estimation," Journal of Statistical Planning and Inference, vol.83, no.2, pp.395–412, 2000.

[40] M. Yamada, M. Sugiyama, G. Wichern, and J. Simm, "Improving the accuracy of least-squares probabilistic classifiers," IEICE Transactions on Information and Systems, vol.E94-D, no.6, pp.1337–1340, 2011.

[41] J. Zhu and E. Hovy, "Active learning for word sense disambiguation with methods for addressing the class imbalance problem," Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp.783–790, 2007.