Active Learning Using Pre-clustering

Hieu T. Nguyen TAT@SCIENCE.UVA.NL
Arnold Smeulders SMEULDERS@SCIENCE.UVA.NL
Intelligent Sensory Information Systems, University of Amsterdam, Faculty of Science, Kruislaan 403, NL-1098 SJ, Amsterdam, The Netherlands

Abstract

The paper is concerned with two-class active learning. While the common approach for collecting data in active learning is to select samples close to the classification boundary, better performance can be achieved by taking into account the prior data distribution. The main contribution of the paper is a formal framework that incorporates clustering into active learning. The algorithm first constructs a classifier on the set of the cluster representatives, and then propagates the classification decision to the other samples via a local noise model. The proposed model allows to select the most representative samples as well as to avoid repeatedly labeling samples in the same cluster. During the active learning process, the clustering is adjusted using the coarse-to-fine strategy in order to balance between the advantage of large clusters and the accuracy of the data representation. The results of experiments in image databases show a better performance of our algorithm compared to the current methods.

1. Introduction

In recent years, research interest has been attracted to semi-supervised learning or learning in the condition that only a small initial amount of data is labeled while the majority of the data remain unlabeled. While many methods focus to improve the supervised learning by using the information from unlabeled data (Seeger, 2001), another important topic is a good strategy in selecting the data to label, considering that labeling data is a time-consuming job. The topic is known as *active learning* (Lewis & Gale, 1994).

Consider the problem of learning a binary classifier on a partially labeled database $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$. Let $\mathcal{D}_\ell$ be the labeled set in which every sample is given a label $y \in \{1, -1\}$, and $\mathcal{D}_u = \mathcal{D} \setminus \mathcal{D}_\ell$. The active learning system comprises two parts: a learning engine and a selection engine. At every iteration the learning engine uses a supervised learning algorithm to train a classifier on $\mathcal{D}_\ell$. The selection engine then selects a sample from $\mathcal{D}_u$ and requests a human expert to label the sample before passing it to the learning engine. The major goal is to achieve a good classifier as best as possible within a reasonable number of calls for labeling by human help.

Current methods on active learning can be characterized by their base learning algorithms which include probabilistic naive Bayes (Nigam et al., 2000; Roy & McCallum, 2001), combination of naive Bayes and logistic regression (Lewis & Gale, 1994), and the Support Vector Machine (SVM) (Campbell et al., 2000; Tong & Koller, 2001; Schohn & Cohn, 2000). The naive Bayes classifier suffers from two problems. First, the classifier assumes the independence between the component features of $\boldsymbol{x}$. This assumption is often violated. The second problem is that naive Bayes is a generative model for which training relies on the estimation of the likelihood $p(\boldsymbol{x}|y)$. This estimation is inaccurate in the case of active learning since the training data are not randomly collected. The paper focuses on discriminative models including logistic regression and SVM. These models aim to estimate the posterior probability $p(y|\boldsymbol{x})$. They are less sensitive to the way the training data is collected, and hence, are more suitable for active learning. A more theoretical consideration is given in (Zhang & Oles, 2000).

It is crucial to choose the most "valuable" training samples. Many methods choose the most uncertain samples which are closest to the current classification boundary. We name this approach the closest-to-boundary criterion. This simple and intuitive criterion performs well in some applications (Lewis & Gale, 1994; Tong & Chang, 2001; Schohn & Cohn, 2000; Campbell et al., 2000). Some other criteria have been proposed specifically for SVM. In (Campbell et al., 2000), it is proposed to select the sample that yields the largest decrease of the margin between the two classes. The method of (Tong & Koller, 2001) selects the

Appearing in *Proceedings of the 21^{st} International Conference on Machine Learning*, Banff, Canada, 2004.

sample that halves the permitted region of the SVM parameters in the parameter space. Both (Campbell et al., 2000) and (Tong & Koller, 2001) need to predict the values of the SVM parameters for every possible case where a candidate sample might be added to the training set. Since it is hard to do this efficiently, the references finally resort to the closest-to-boundary criterion.

The closest-to-boundary methods ignore the prior data distribution which can be useful for active learning. In (Cohn et al., 1996), it is suggested to select samples that minimize the expected future classification error:

$$\int_{\boldsymbol{x}} E\left[(\hat{y}(\boldsymbol{x}) - y(\boldsymbol{x}))^2 | \boldsymbol{x}\right] p(\boldsymbol{x}) d\boldsymbol{x} \quad (1)$$

where $y(\boldsymbol{x})$ is the true label of $\boldsymbol{x}$ and $\hat{y}(\boldsymbol{x})$ is the classifier output. $E[.|\boldsymbol{x}]$ denotes the expectation over $p(y|\boldsymbol{x})$. Due to the complexity of the integral, the direct implementation of eq. (1) is usually difficult. However, it shows that the data uncertainty should be weighted with the prior density $p(\boldsymbol{x})$. If $p(\boldsymbol{x})$ is uniform or unknown, the expectation under the integral is the contribution by a sample into the classification error. The expectation can then be used to measure the value of the sample in the condition that the computation of the integral is complex. Under the assumption that the current classification boundary is good, it is easy to show that the error expectation is maximal for the samples lying on the classification boundary, see section 3.3. When $p(\boldsymbol{x})$ is known and non-uniform, the information about the distribution can be used to select better data. In this paper $p(\boldsymbol{x})$ is obtained via clustering which can be done offline without the interaction with human. The clustering information is then useful for active learning in two ways. First, the representative samples located in center of clusters are more important than the other, and should be selected first in labeling. Secondly, samples in the same cluster are likely to have the same label, (Seeger, 2001; Chapelle et al., 2002). This assumption should be used to accelerate active learning by reducing the number of labeling samples from the same cluster.

The idea to combine clustering and active learning has appeared in previous work. In (McCallum & Nigam, 1998), a naive Bayes classifier is trained over both labeled and unlabeled data using an EM algorithm. Under the condition that the overwhelming majority of the data is unlabeled, that training algorithm amounts to clustering the data set, and the role of the labeled data is for initialization only. Clustering information also contributes to the selection where an uncertainty measure is weighted with the density of the sample. The referenced approach does not match, however, the objective of this paper to combine clustering with a discriminative model. Several other active learning schemes also weigh the uncertainty with the data density (Zhang & Chen, 2002; Tang et al., 2002). Some methods put more emphasis on the sample representativeness by selecting cluster centers from a set of most interesting samples. In the representative sampling by (Xu et al., 2003), the algorithm uses the k-means algorithm to cluster the samples lying within the margin of a SVM classifier trained on the current labeled set. The samples at cluster centers are then selected for human labeling. The method of (Shen & Zhai, 2003) has a similar idea, but applies the k-medoid algorithm for the top relevant samples. In general, heuristic methods have been proposed to balance between the uncertainty and the representativeness of the selected sample. They encourage the selection of cluster centers. However, no measure has been taken to avoid repeatedly labeling samples in same cluster. In addition, there are important questions that remain open, namely, how to adapt the classification model for a training set that contains only cluster centers? and, how to classify samples that are disputed by several clusters? This paper presents a solution for these issues using a mathematical model that explicitly takes clustering into account.

The organization of the paper is as follows. Section 2 describes the incorporation of the clustering information into the data model, and provides the theoretical framework for the data classification. Section 3 presents our active learning algorithm. Section 4 shows the results of the algorithm for the classification of images in test databases.

2. Probabilistic framework

2.1. Data model

In the standard classification, data generation is described by the joint distribution $p(\boldsymbol{x}, y)$ of the data $\boldsymbol{x}$ and the class label $y \in \{-1, +1\}$. The clustering information is explicitly incorporated by introducing the hidden cluster label $k \in \{1, \ldots, K\}$, where K is the number of clusters in the data. k indicates that the sample belongs to the k-th cluster. Assume that all information about the class label y is already encoded in the cluster label k. This implies that once k is known, y and $\boldsymbol{x}$ are independent. The joint distribution is written as:

$$p(\boldsymbol{x}, y, k) = p(y|\boldsymbol{x}, k)p(\boldsymbol{x}|k)p(k) = p(y|k)p(\boldsymbol{x}|k)p(k) \quad (2)$$

The simple Bayesian belief net representing the model is depicted in Figure 1.

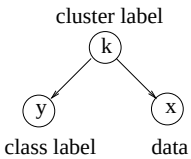

Figure 1. The Bayesian net for the data model.

Before giving the specific form for the three distributions in eq. (2) we remark that a similar scheme has been proposed for the passive semi-supervised learning (Miller & Uyar, 1996; Seeger, 2001). The conceptual difference, however, between their approach and ours is in the definition of $p(y|k)$. In the references, $p(y|k)$ is defined within individual clusters. As a consequence, the estimation of the parameters of $p(y|k)$ can be unreliable due to insufficient labeled data in a cluster. In our model, $p(y|k)$ is defined for all clusters with the same parameters.

We use logistic regression for $p(y|k)$:

$$p(y|k) = \frac{1}{1 + \exp\{-y(\boldsymbol{c}_k.\boldsymbol{a} + b)\}} \tag{3}$$

Here, $\boldsymbol{c}_k$ is a representative of the k-th cluster which is determined via clustering. $\boldsymbol{a} \in \mathbb{R}^d$ and $b \in R$ are the logistic regression parameters. In essence, $p(y|k)$ is the label model for a representative subset of the database.

In the ideal case where data is well clustered, once all the parameters of $p(y|k)$ are determined, one could use this probability to determine the label of the cluster representatives, and then assign the same label to the remaining samples in the cluster. In practice, however, clustering can be inaccurate and we will have problems with classification of samples at border between the clusters. To achieve better classification for those samples, we use a soft cluster membership which allows a sample to be connected to more than one clusters (representatives) with a probability. The noise distribution $p(\boldsymbol{x}|k)$ is then used to propagate information of label y from the representatives into the remaining majority of the data, see Figure 2. We use the isotropic Gaussian model:

$$p(\boldsymbol{x}|k) = (2\pi)^{-d/2}\sigma^{-d}\exp\{-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{c}_k\|^2\} \tag{4}$$

where σ^2 is the variance assumed to be the same for all clusters.

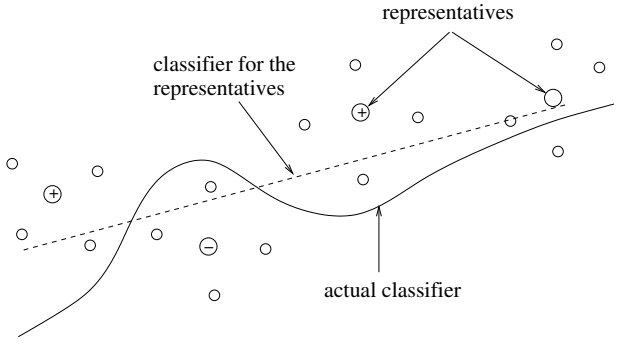

Figure 2. The classification model

Let $p(k) = \alpha_k$. Then, $p(\boldsymbol{x})$ is a mixture of K Gaussians with the weights α_k.

In the presented method, the parameters $\boldsymbol{c}_k$, α_k, $\boldsymbol{a}$ and b are estimated from the data. The scale parameter σ^2 is given initially. It can be changed during active learning when a different clustering setting is needed.

2.2. Data classification

Given the above model, one calculates $p(y|\boldsymbol{x})$, the posterior probability of label of a sample as follows:

$$p(y|\boldsymbol{x}) = \sum_{k=1}^{K} p(y,k|\boldsymbol{x}) = \sum_{k=1}^{K} p(y|k)p(k|\boldsymbol{x}) \tag{5}$$

where $p(k|\boldsymbol{x}) = p(\boldsymbol{x}|k)p(k)/p(\boldsymbol{x})$.

Data are then classified using the Bayes decision rule:

$$\hat{y}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } p(y=1|\boldsymbol{x};\hat{\boldsymbol{a}},\hat{b}) > p(y=-1|\boldsymbol{x};\hat{\boldsymbol{a}},\hat{b}) \\ -1 & \text{otherwise} \end{cases} \tag{6}$$

where $\hat{\boldsymbol{a}}, \hat{b}$ denote the current estimates of the parameters.

Observe from eq. (5) that the classification decision is a weighted combination of the classification decision for the representatives. Well clustered samples will be assigned the same label as the nearest representative. Samples disputed by several clusters, on the other hand, will be assigned the label of the cluster, which has the highest confidence. Note that the weights $p(k|\boldsymbol{x})$ are fixed unless the data are re-clustered whereas $p(y|k)$ is updated upon the arrival of new training data.

3. Description of algorithm

The parameters of the model proposed in section 2.1 are estimated via likelihood maximization. The data likelihood comprises two parts: the likelihood of the labeled data and the likelihood of the unlabeled data:

$$L = \sum_{i\in\mathcal{I}_\ell} \ln p(\boldsymbol{x}_i, y_i) + \sum_{i\in\mathcal{I}_u} \ln p(\boldsymbol{x}_i) \tag{7}$$

where $\mathcal{I}_\ell$ and $\mathcal{I}_u$ denote the set of indices of labeled and unlabeled samples respectively. Expanding $\ln p(\boldsymbol{x}_i, y_i)$ as the sum of $\ln p(\boldsymbol{x}_i)$ and $\ln p(y_i|\boldsymbol{x}_i)$, the likelihood (7) can be written with explicit dependence on the parameters as follows:

$$\begin{aligned} L(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K,\alpha_1,\ldots,\alpha_K,\boldsymbol{a},b) = \\ \sum_{i\in\mathcal{I}_\ell\cup\mathcal{I}_u} \ln p(\boldsymbol{x}_i;\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K,\alpha_1,\ldots,\alpha_K) + \\ \sum_{i\in\mathcal{I}_\ell} \ln p(y_i|\boldsymbol{x}_i;\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K,\boldsymbol{a},b) \end{aligned} \tag{8}$$

As the amount of the unlabeled data is overwhelming over the labeled data, the parameters $\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K$ and

$\alpha_1, \ldots, \alpha_K$ are determined mainly by maximizing the first term in eq. (8). The maximization of each term can therefore be done separately. The clustering algorithm maximizes the likelihood of the data samples to obtain the cluster representatives and the cluster prior. The maximization of the label likelihood follows to estimate the parameters $\boldsymbol{a}$ and b.

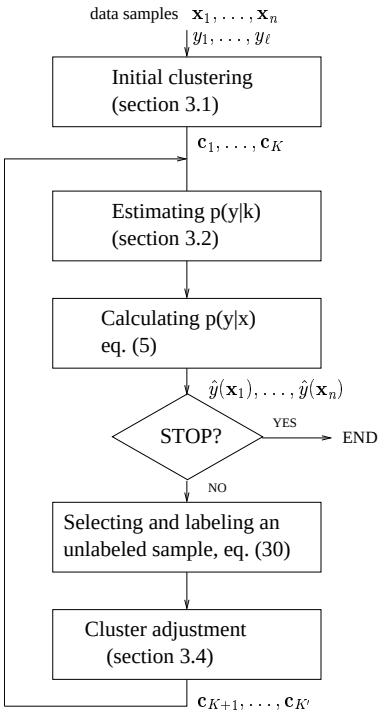

Figure 3. The proposed active learning algorithm.

The block scheme of the algorithm is illustrated in Figure 3.

3.1. Initial clustering

In the presented algorithm, the goal of clustering is data representation rather than data classification. We therefore use the K-medoid algorithm of (Kaufman & Rousseeuw, 1990). The algorithm finds K representatives $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$ of the data set $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ so as to minimize the sum of the distance from the data samples to the nearest representative. See (Struyf et al., 1997) for the detailed implementation.

The K-medoid algorithm is computationally expensive when either n or K is large. In practical applications both numbers are very large indeed. The following simplifications are employed to reduce computations. First, the data set is split into smaller subsets. The K-medoid algorithm is then applied to cluster every subset into a limited number K_0 of clusters. Clustering is continued by subsequently breaking the cluster with the largest radius r_k into two smaller ones with:

$$r_k = \max_{i \in \mathcal{I}_k} |\boldsymbol{x}_i - \boldsymbol{c}_k| \tag{9}$$

where $\mathcal{I}_k$ denotes the set of indices of the samples in the k-th cluster. The process of cluster fission is completed when:

$$\max_k r_k < \kappa\sigma \tag{10}$$

where κ is a predefined constant. We have used $\kappa = 8$. Thus, the cluster size and the final number of clusters K is controlled by the scale parameter σ.

Once the cluster representatives $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$ have been determined, the cluster prior α_k is obtained by iterating the following two equations until stability:

$$p(k|\boldsymbol{x}_i) = \frac{\alpha_k \exp\{-\frac{1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{c}_k\|^2\}}{\sum_{k'=1}^{K} \alpha_{k'} \exp\{-\frac{1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{c}_{k'}\|^2\}} \tag{11}$$

$$\alpha_k = \frac{1}{n}\sum_{i=1}^{n} p(k|\boldsymbol{x}_i) \tag{12}$$

3.2. The estimation of the class label model

This section presents the estimation of the distribution $p(y|k)$ based on the maximization of the second likelihood in eq. (8). Fixing the cluster representatives $\boldsymbol{c}_k$, the likelihood depends only on the parameters $\boldsymbol{a}$ and b:

$$\max_{\boldsymbol{a},b} \sum_{i \in \mathcal{I}_\ell} \ln p(y_i|\boldsymbol{x}_i; \boldsymbol{a}, b) \tag{13}$$

From eq.(5), $p(y_i|\boldsymbol{x}_i; \boldsymbol{a}, b)$ can be written as a mixture of K logistic distributions $p(y_i|k; \boldsymbol{a}, b)$ with the weights $p(k|\boldsymbol{x}_i)$.

In case the dimensionality d is higher than the number of the labeled samples, the optimization in (13) is numerically unstable. The conventional approach to overcome the problem is to add a regularization term $\frac{1}{2}\lambda\|\boldsymbol{a}\|^2$ where λ is a predefined parameter. This leads to the minimization of the following objective function:

$$\mathcal{L}(\boldsymbol{a}, b) = \frac{1}{2}\lambda\|\boldsymbol{a}\|^2 - \sum_{i \in \mathcal{I}_\ell} \ln\left\{\sum_{k=1}^{K} p(k|\boldsymbol{x}_i) p(y_i|k; \boldsymbol{a}, b)\right\} \tag{14}$$

Remark that eq. (14) is the extension of the regularized logistic regression (Zhang & Oles, 2001; Zhu & Hastie, 2001) for the mixture of the logistic distributions.

The minimization of $\mathcal{L}$ is implemented using Newton's algorithm which guarantees to find a local minimum. Furthermore, since $\mathcal{L}$ is convex, it has only one local minimum which is also the global minimum.

Starting with an initial guess $\boldsymbol{a}_0$ and b_0, the parameters $\boldsymbol{a}$ and b are updated iteratively. At each iteration, the param-

eter increment in the steepest direction is:

$$\begin{bmatrix}\Delta \boldsymbol{a} \\ \Delta b\end{bmatrix} = -\boldsymbol{H}^{-1}\nabla\mathcal{L} \tag{15}$$

where $\nabla\mathcal{L}$ is the Jacobian of $\mathcal{L}$, and $\boldsymbol{H}$ is a positive definite approximation of the hessian matrix of $\mathcal{L}$. Using eq.(3), it can be shown that:

$$\nabla\mathcal{L} = \begin{bmatrix}\frac{\partial\mathcal{L}}{\partial \boldsymbol{a}} \\ \frac{\partial\mathcal{L}}{\partial b}\end{bmatrix} = \lambda\begin{bmatrix}\boldsymbol{a} \\ 0\end{bmatrix} - \sum_{k=1}^{K}\gamma_k\begin{bmatrix}\boldsymbol{c}_k \\ 1\end{bmatrix} \tag{16}$$

where

$$\gamma_k = \sum_{i\in\mathcal{I}_\ell} y_i p(y_i|k;\boldsymbol{a},b)[1-p(y_i|k;\boldsymbol{a},b)]\frac{p(k|\boldsymbol{x}_i)}{p(y_i|\boldsymbol{x}_i;\boldsymbol{a},b)} \tag{17}$$

For the hessian matrix, the following approximation can be used:

$$\boldsymbol{H} = \lambda\begin{bmatrix}\boldsymbol{I} & 0 \\ 0 & 0\end{bmatrix} + \sum_{k=1}^{K}\tau_k\begin{bmatrix}\boldsymbol{c}_k\boldsymbol{c}_k^T & \boldsymbol{c}_k \\ \boldsymbol{c}_k^T & 1\end{bmatrix} \tag{18}$$

where $\boldsymbol{I}$ is the identity matrix and:

$$\tau_k = \sum_{i\in\mathcal{I}_\ell} p(y_i|k;\boldsymbol{a},b)^2[1-p(y_i|k;\boldsymbol{a},b)]\frac{p(k|\boldsymbol{x}_i)}{p(y_i|\boldsymbol{x}_i;\boldsymbol{a},b)} \tag{19}$$

To get more insight into eq.(15), let:

$$\begin{aligned} \boldsymbol{z} &= \sum_{k=1}^{K}\tau_k\boldsymbol{c}_k / \sum_{k=1}^{K}\tau_k & (20)\\ D &= [\sqrt{\tau_1}\boldsymbol{c}_1,\ldots,\sqrt{\tau_K}\boldsymbol{c}_K] & (21)\\ G &= DD^T + \lambda\boldsymbol{I} & (22)\end{aligned}$$

Here, D is the $d\times K$ matrix whose columns are the vectors $\sqrt{\tau_k}\boldsymbol{c}_k$. One can show that:

$$\begin{aligned} \Delta\boldsymbol{a} &= -G^{-1}\left[\frac{\partial\mathcal{L}}{\partial\boldsymbol{a}} - \boldsymbol{z}\frac{\partial\mathcal{L}}{\partial b}\right] & (23)\\ \Delta b &= \boldsymbol{z}^T G^{-1}\left[\frac{\partial\mathcal{L}}{\partial\boldsymbol{a}} - \boldsymbol{z}\frac{\partial\mathcal{L}}{\partial b}\right] - \frac{1}{\sum_{k=1}^{K}\tau_k}\frac{\partial\mathcal{L}}{\partial b} & (24)\end{aligned}$$

If d is high, it is efficient to invert G using the Woodbury formula:

$$G^{-1} = \frac{1}{\lambda}[\boldsymbol{I} - D(\lambda\boldsymbol{I} + D^T D)^{-1}D^T] \tag{25}$$

If all τ_k are non-zero, the size of $D^T D$ is $K\times K$. Since K is large, the inverting of G as in eq. (25) would be computationally expensive still. However, remark that for a sample $\boldsymbol{x}_i$ there are only few values of k such that $p(k|\boldsymbol{x}_i)$ is different from zero, especially if $\boldsymbol{x}_i$ is a cluster representative. In the latter case, the sample typically belongs to one cluster only. As will be seen in the next subsection, the presented algorithm tends to select the training data from the cluster representatives. The number of non-zero τ_k is then small, approximately the same as the number of labeled samples. The computation of G^{-1} in eq. (25) can then be done efficiently by suppressing the columns in D which correspond to τ_k that equal to zero.

3.3. Criterion for data selection

The selection criterion gives priority to two types of samples: samples close to the classification boundary and samples which are cluster representatives. Furthermore, within the set of cluster representatives, one should start with the highest density clusters first.

We have noted that the computation of the future classification error in eq.(1) is complicated. So, instead of choosing the sample that produces the smallest future error, we select the sample that has the largest contribution to the current error. Although such approach does not guarantee the smallest future error, there is a good chance for a large decrease of the error. The selection criterion is:

$$s = \arg\max_{i\in\mathcal{I}_u} E\left[(\hat{y}_i - y_i)^2|\boldsymbol{x}_i\right]p(\boldsymbol{x}_i) \tag{26}$$

where s denotes the index of the selected sample.

The error expectation for an unlabeled $\boldsymbol{x}_i$ is calculated over the distribution $p(y_i|\boldsymbol{x}_i)$:

$$\begin{aligned} &E\left[(\hat{y}_i - y_i)^2|\boldsymbol{x}_i\right] \\ &\quad= p(y_i=1|\boldsymbol{x}_i)(\hat{y}_i-1)^2 + p(y_i=-1|\boldsymbol{x}_i)(\hat{y}_i+1)^2 \\ &\quad= 2 - 2\hat{y}_i(p(y_i=1|\boldsymbol{x}_i) - p(y_i=-1|\boldsymbol{x}_i)) \end{aligned} \tag{27}$$

It should be noted that the probability $p(y_i=1|\boldsymbol{x}_i)$ is unknown and needs to be approximated. An obvious choice is to use the current estimation $p(y_i=1|\boldsymbol{x}_i;\hat{\boldsymbol{a}},\hat{b})$, assuming $\hat{\boldsymbol{a}},\hat{b}$ are good enough. Letting

$$f(\boldsymbol{x}_i) = p(y_i=1|\boldsymbol{x}_i;\hat{\boldsymbol{a}},\hat{b}) - p(y_i=-1|\boldsymbol{x}_i;\hat{\boldsymbol{a}},\hat{b}) \tag{28}$$

it follows from eq. (5) that:

$$E\left[(\hat{y}_i - y_i)^2|\boldsymbol{x}_i\right] = 2(1-|f(\boldsymbol{x}_i)|) \tag{29}$$

Observe that if $\boldsymbol{x}_i$ lies on the current classification boundary, the quantity $|f(\boldsymbol{x}_i)|$ is minimal, and hence the expected error is maximal.

Eq. (26) becomes:

$$s = \arg\max_{i\in\mathcal{I}_u}(1-|f(\boldsymbol{x}_i)|)p(\boldsymbol{x}_i) \tag{30}$$

where

$$p(\boldsymbol{x}_i) = \sum_{k=1}^{K}\alpha_k\exp\{-\frac{1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{c}_k\|^2\} \tag{31}$$

Figure 4. Example view of images in the first database.

The resulting criterion indeed satisfies the demands put in the beginning of the subsection. The term $(1 - |f(\boldsymbol{x}_i)|)$ gives priority to the samples at the boundary. Meantime, $p(\boldsymbol{x}_i)$ gives priority to the representatives of dense clusters.

3.4. Coarse-to-fine adjustment of clustering

The labeling of high density clusters promises a substantial move of the classification boundary. It is therefore advantageous to group the data into large clusters in the initial clustering. This is achieved by setting a high value for the initial scale parameter σ_0. When the classification boundary reaches the border between the global clusters, a finer clustering with smaller cluster size is better to obtain a more accurate classification boundary. The maximum in eq. (30) can be used as the indication for the need to adjust the clustering. If this quantity drops below a threshold t_e:

$$\max_{i \in \mathcal{I}_u} (1 - |f(\boldsymbol{x}_i)|) p(\boldsymbol{x}_i) < t_e \tag{32}$$

the scale parameter is decreased:

$$\sigma_{new} = \eta \sigma_{old} \tag{33}$$

where $0 < \eta < 1$. The data set is then re-clustered. The parameters t_e and η are predefined. We have used $t_e = 0.2$ and $\eta = 0.9$. Note that clustering the data set with different scales can be done offline. Furthermore, change of the scale takes place not in every iteration, but only few times during the learning process.

4. Experiments

We have performed two experiments to test the performance of the proposed algorithm. In the first experiment, the algorithm is applied to find human face images in a database containing 2500 images of size 20×20. See (Pham et al., 2002) for details on how the images were created. Example views of some images are shown in Figure 4. In the second experiment, a test database was made of images of handwritten digits taken from the MNIST database (http://yann.lecun.com/exdb/mnist/). The size of images is 28×28. The objective is to separate the images of a given digit against the other nine.

In the experiments the following setting was used. The images are considered as the vectors composed of the pixel grey values which range from 0 to 255. The initial training set contains equal numbers of object and non-object images. The initial size of this set was $0.4\%n$ and was increased to $3.5\%n$ during active learning, where n is the number of samples in the database. For clustering, the databases were split into subsets per 1250 samples. The K-medoid algorithm was applied for each subset with the initial number of clusters $K_0 = 20$. The initial value of σ was $\sigma_0 = 2.2d$ where d is the number of pixels in one image. For the estimation of the class label distribution, we use the regularization coefficient $\lambda = 80$.

Every time a new training sample is added, the classifier is re-trained and tested on the rest of the database. The classification error is calculated as the sum of the missed positives and false alarms relative to n. The performance evaluation is based on the decrease of the classification error as the function of the amount of training samples.

For comparison, we have also implemented three other active learning algorithms. They use the standard linear SVM for classification. The first algorithm selects training data randomly. In the second algorithm, data are selected according to the closest-to-boundary criterion. The third algorithm uses the representative sampling of (Xu et al., 2003) which selects the medoid centers in the SVM margin. As we select one sample per iteration, this leads to selection of the most representative sample among the samples in the margin.

Figure 5 shows the result of the first experiment for different proportions between the numbers of face and non-face images in the database. Figure 6 shows the result of the second experiment for the different sizes of the database. Both figures show the average of the classification error obtained by repeating the experiments with three different initial training sets that are picked up randomly. The results of Figure 6 are also an average over the ten digits.

The proposed algorithm outperforms all three other algorithms. The most significant improvement is observed in Figure 5a with equal numbers for object and non-object samples in the database. The improvement decreases when the amount of the object samples is small relative to the non-object samples, see Figure 5c. In this case, since there are no clusters of the object class, the proposed algorithm is not advantageous over the closest-to-boundary algorithm in finding object samples. Nevertheless, the proposed algorithm remains better as it still benefits from the clustering of non-object samples. Representative sampling turns out

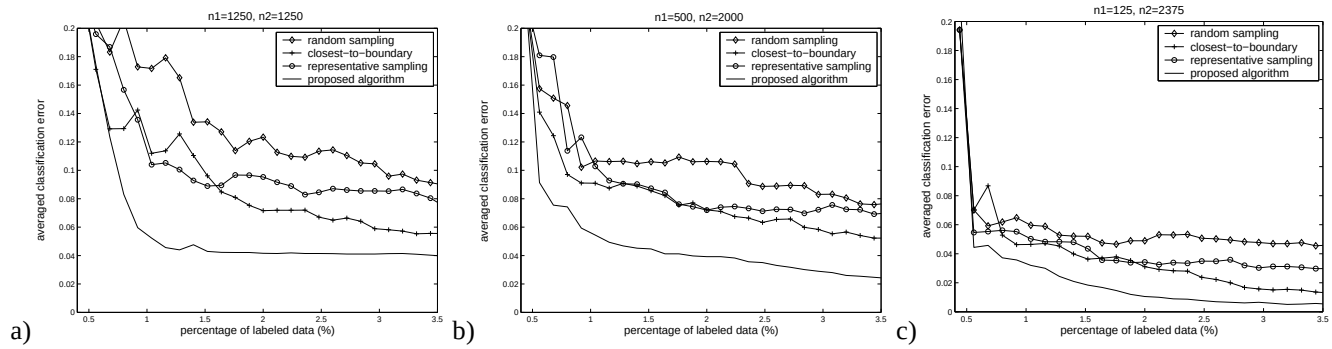

Figure 5. The results for the classification of face images. n_1 and n_2 are the number of face and non-face images in the databases respectively.

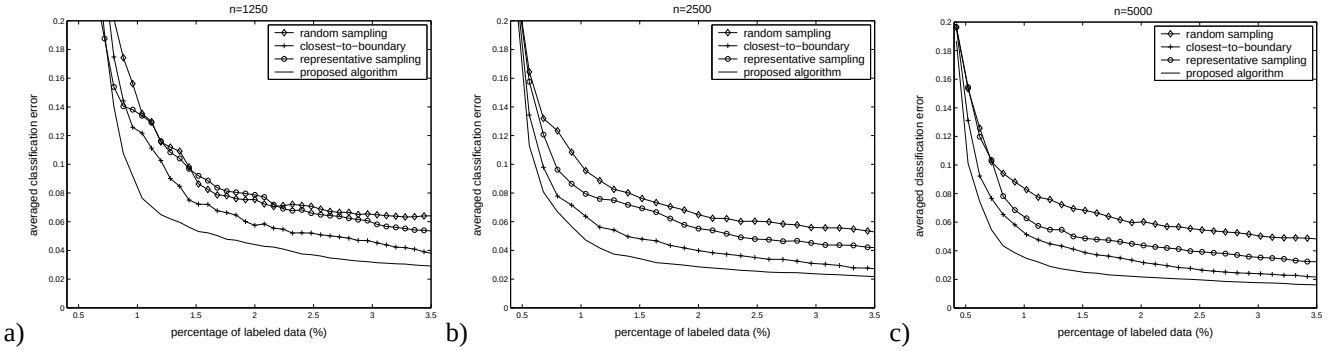

Figure 6. The classification results of handwritten digits from the MNIST database. n is the database size.

to perform better only than random sampling. A possible reason could be the undervaluation of the uncertainty and the lack of a proper classification model.

5. Conclusion

The paper has proposed a formal model for incorporation of clustering into active learning. The model allows to select most representative training examples as well as to avoid repeatedly labeling samples in same cluster, leading to better performance than the current methods. To take the advantage of the similarity between the class label of data in the same cluster, the method first constructs a classifier over the population of the cluster representatives. We use regularized logistic regression which is a discriminative model with state-of-the-art performance and which is naturally fitted into a probabilistic framework. The gaussian noise model is then used to infer the class label for non-representative samples. New training data are selected from the samples having the maximal contribution to the current expected error. In addition to closeness to the classification boundary, the selection criterion gives priority also to the representatives of the dense clusters, making the training set statistically stable.

The method was restricted to linear logistic regression as the main purpose of the paper is to show the advantage of using clustering information. We have succeeded in that goals for the given datasets.

References

Campbell, C., Cristianini, N., & Smola, A. (2000). Query learning with large margin classifiers. *Proc. 17th International Conf. on Machine Learning* (pp. 111–118). Morgan Kaufmann, CA.

Chapelle, O., Weston, J., & Scholkopf, B. (2002). Cluster kernels for semi-supervised learning. *Advances in Neural Information Processing Systems*.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence research, 4*, 129–145.

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis.* John Wiley & Sons.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. *Proceedings of SIGIR-94, 17th ACM International Conference on Research*

and Development in Information Retrieval (pp. 3–12). Springer Verlag.

McCallum, A. K., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. *Proc. 15th International Conf. on Machine Learning* (pp. 350–358). Morgan Kaufmann, CA.

Miller, D., & Uyar, H. (1996). A mixture of experts classifier with learning based on both labelled and unlabelled data. *Advances in Neural Information Processing Systems 9* (pp. 571–577).

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*, 103–134.

Pham, T., Worring, M., & Smeulders, A. (2002). Face detection by aggregated bayesian network classifiers. *Pattern Recogn. Letters, 23*, 451–461.

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. *Proc. 18th International Conf. on Machine Learning* (pp. 441–448). Morgan Kaufmann, CA.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines. *Proc. 17th International Conf. on Machine Learning* (pp. 839–846). Morgan Kaufmann, CA.

Seeger, M. (2001). *Learning with labeled and unlabeled data* (Technical Report). Edinburgh University.

Shen, X., & Zhai, C. (2003). Active feedback - UIUC TREC-2003 HARD experiments. *The 12th Text Retrieval Conference, TREC*.

Struyf, A., Hubert, M., & Rousseeuw, P. (1997). Integrating robust clustering techniques in s-plus. *Computational Statistics and Data Analysis, 26*, 17–37.

Tang, M., Luo, X., & Roukos, S. (2002). Active learning for statistical natural language parsing. *Proc. of the Association for Computational Linguistics 40th Anniversary Meeting*. Philadelphia, PA.

Tong, S., & Chang, E. (2001). Support vector machine active learning for image retrieval. *Proceedings of the 9th ACM int. conf. on Multimedia* (pp. 107–118). Ottawa.

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research, 2*, 45–66.

Xu, Z., Yu, K., Tresp, V., Xu, X., & Wang, J. (2003). Representative sampling for text classification using support vector machines. *25th European Conf. on Information Retrieval Research, ECIR 2003*. Springer.

Zhang, C., & Chen, T. (2002). An active learning framework for content-based information retrieval. *IEEE trans on multimedia, 4*, 260–268.

Zhang, T., & Oles, F. (2000). A probability analysis on the value of unlabeled data for classification problems. *Proc. Int. Conf. on Machine Learning*.

Zhang, T., & Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval, 4*, 5–31.

Zhu, J., & Hastie, T. (2001). Kernel logistic regression and the import vector machine. *Advances in Neural Information Processing Systems*.