数理解析研究所講究録
第1584巻 2008年 230-244

Active Learning for Semi-Supervised Kernel Logistic Regression

Yasutoshi Yajima
Tokyo Institute of Technology

Teppei Sato
Cybozu, Inc.

Abstract

The purpose of this paper is to introduce new approaches for kernel logistic regression in a semi-supervised setting. Using the special structure of Laplacian kernel matrices, we propose new formulations which minimize the negative log likelihood of the kernel logistic regression model quite efficiently. The proposed formulations can be seen as a natural extension of the recently developed semi-supervised learning method based on Gaussian random field and harmonic functions.

Also, we propose new algorithms for performing pool-based active learning based on A-optimality in which the semi-supervised kernel logistic regression is used to estimate the class probabilities. We show that the active learning algorithms can be carried out in the feature space defined by the associated kernel matrices. We give experimental results showing that the proposed active learning method can generate accurate classifiers using a fewer number of labeled data points compared with the random queries.

1 Introduction

In the supervised learning framework, we first need to prepare a number of labeled data points as a set of training data points. In many practical situations such as text classification or Web search, however, the task for obtaining labeled data points is very expensive, while a large number of unlabeled data points are easily available.

Recently, several studies [22, 20] have been devoted to develop *semi-supervised learning* in which unlabeled data points are employed to achieve quite high accuracies for the classification problems. These approaches are based on Gaussian random field and harmonic functions, and often referred to as energy minimization. It has been shown that [18, 17] these approaches can be viewed as a special case of support vector machine (SVM) with kernels defined on a graph [14].

Active learning can also be used to obtain accurate classifiers with a small number of labeled data points. In active learning, the learner actively select a data point as a query from a pool of unlabeled data so as to improve the accuracy of the current classifier. Then, the selected data point is added to the labeled set after its true label is assigned by a human expert, and the classifier is retrained. Several methods have been proposed for selecting good queries from the pool, which include uncertainty sampling [9], query by committee [12], and the version space minimization [15]. Recently, Zhu *et al.* [23] introduced a new approach which combines semi-supervised learning and active learning.

In this paper, we also propose semi-supervised learning which can be actively performed in a different way than [23]. We first introduce formulations for semi-supervised logistic regression with kernels defined on a graph, which is a natural extension of the semi-supervised SVM given in [18]. We then introduce a new active learning method based on A-optimality, in which the predicted variance is minimized. The A-optimality criterion is often used for regression analysis in the field of experimental design in statistics, as well as neural networks [10, 2].

The contributions of this work include the following:

(i) New formulations for kernel logistic regression are developed in a semi-supervised setting, which enables us to naturally handle the multiclass classification problems in a semi-supervised setting, while the energy minimization approaches [22, 20] are essentially limited to the binary case.

(ii) Unlike the existing dual approach for kernel logistic regression [8], we can employ the kernel matrices in the primal formulation which minimizes the negative log likelihood. Using the special structure of the Laplacian kernel matrices, we can obtain quite simple formulations which can be minimized efficiently.

(iii) Using the efficient procedure for the kernel logistic regression, we develop a method for calculating the A-optimality criterion for active learning. We show that this active learning method can be carried out in the feature space characterized by the associated kernel matrices.

The remaining part of the paper is organized as follows. In Sect. 2, we briefly review logistic regression and the formulations for its kernel-based extension. Sect. 3 is devoted to develop the semi-supervised setting of the kernel-based logistic regression. In Sect. 4, we introduce a method for calculating the kernel-based A-optimality criterion, which is extended to the multiclass case in Sect. 5. Numerical experiments using a real world data set are presented in Sect. 6, and conclusions are presented in Sect. 7.

2 Kernel Logistic Regression

2.1 Regularized Classifiers

Suppose that we have a set of l training data points denoted by

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}, \tag{2.1}$$

where $x_j \in \mathbf{R}^N$ stands for the j-th training vector with N-dimensional attributes, and $y_j \in \{-1, +1\}$ is the corresponding binary class label.

In logistic regression, we estimate the conditional probability of the label y given x in the form of

$$P(y|x) = \frac{1}{1 + \exp(-y \langle w, x \rangle)}, \tag{2.2}$$

where $w \in \mathbf{R}^N$ is an N-dimensional vector of parameters. The parameter w is learned by maximizing the likelihood $\prod_{j=1}^{l} P(y_j|x_j)$, or equivalently by minimizing the negative log likelihood given below:

$$\sum_{j=1}^{l} \log\left(1 + \exp(-y_j \langle w, x_j \rangle)\right).$$

In order to improve numerical difficulties and to avoid overfitting, we usually consider the regularized logistic regression [19], which is formulated as follows:

$$\left| \text{Min.} \quad \frac{\lambda}{2}\|w\|^2 + \sum_{j=1}^{l} \log\left(1 + \exp(-y_j \langle w, x_j \rangle)\right), \right. \tag{2.3}$$

where λ is a positive parameter controlling the degree of regularization.

The idea of adding the regularization term is also used in several methods, which include linear ridge regression [4] and SVM [16]. In SVM, the following linear discriminating function with a gradient vector w and a threshold b is considered

$$d(x) = \langle w, x \rangle + b.$$

The parameters w and b are learned by solving the following quadratic minimization problem:

$$\left| \text{Min.} \quad \frac{\lambda}{2}\|w\|^2 + \sum_{j=1}^{l} [1 - y_j d(x_j)]_+, \right. \tag{2.4}$$

where $[\cdot]_+$ denotes the hinge loss function:

$$[z]_+ = \begin{cases} z & \text{if } z \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Here, it is worth mentioning that the formulation of SVM given in (2.4) is quite similar to that of regularized kernel logistic regression (2.3). However, the objective function of SVM is not differentiable, while that of the logistic regression is smooth.

In the conventional SVM, instead of optimizing (2.4), we solve the corresponding dual formulation. Let $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_l)^T \in \mathbf{R}^l$ be a vector of dual variables, the dual of (2.4) is given as follows:

$$(2.5) \quad \left| \begin{array}{ll} \text{Min.} & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \langle x_i, x_j \rangle y_i y_j \alpha_i \alpha_j - \sum_{j=1}^{l} \alpha_j \\ \text{s.t.} & \sum_{j=1}^{l} y_j \alpha_j = 0, \\ & 0 \le \alpha_j \le \lambda^{-1}, \ j = 1, 2, \ldots, l. \end{array} \right.$$

Let $\alpha^* = (\alpha_1^*, \cdots, \alpha_l^*)^T$ be an optimal solution of this problem. It has been shown that the associated primal optimal solution w^* is given as a linear combination of the data points, i.e., $w^* = \sum_{j=1}^{l} \alpha_j^* y_j x_j$. See, e.g. [13] for details.

2.2 Kernel Based Nonlinear Classifiers

In the conventional SVM framework, nonlinear classifiers can be generated by transforming the training data points into a high dimensional space. Let us assume that we have a function $\phi(\cdot) : \mathbf{R}^N \mapsto \mathcal{F}$ which nonlinearly maps the original data points into a high dimensional *feature space* $\mathcal{F}$. Generating linear discriminators in the feature space amounts to obtaining nonlinear discriminators in the original data space. In the rest of this paper, let us denote the mapped image $\phi(x_j)$ by ϕ_j, for simplicity.

The dual formulation plays a crucial role for handling the above transformation. It is obvious to see that the dual formulation is defined only by the inner products of the data points. In addition, a number of *kernel functions* have been developed, which give the inner products directly from the original data points, without knowing the transformation $\phi(\cdot)$, explicitly. For example, let t be a hyper-parameter, the following RBF function:

$$\mathcal{K}(x, x') = \exp\left(\frac{-\|x - x'\|^2}{t^2}\right)$$

is one of the most popular kernel functions.

Now, let K be an $l \times l$ symmetric matrix the $i - j$ element of which is the inner product of the mapped images ϕ_i and ϕ_j, i.e., $K_{ij} = \langle \phi_i, \phi_j \rangle$ given by the kernel function. The matrix K is referred to as the kernel matrix. Therefore, replacing the inner product terms in the dual formulation by the kernel functions, we can obtain nonlinear classifiers without knowing ϕ_j's.

Also, let $y^T = (y_1, \ldots, y_l)$ and e be a vector of all ones. Then, let us write the dual problem as follows:

$$(2.6) \quad \left| \begin{array}{ll} \text{Min.} & \frac{1}{2}\alpha^T K \alpha - e^T \alpha \\ \text{s.t.} & y^T \alpha = 0, \\ & 0 \le \alpha_j \le \lambda^{-1}, \ j = 1, 2, \ldots, l. \end{array} \right.$$

We note that this problem is a quite simple convex minimization problem since the kernel function generates a positive semidefinite kernel matrix K, and that, exploiting the special structure, there exist algorithms [7, 11] for solving this problem with a large number of training data points, efficiently.

In logistic regression, employing the same idea of the feature space mapping, we can introduce the kernel based nonlinearity. More precisely, let us first consider the following problem defined in the feature space:

$$(2.7) \quad \left| \begin{array}{ll} \text{Min.} & \frac{\lambda}{2}\|\omega\|^2 + \sum_{j=1}^{l} \log\left(1 + \exp(-y_j \langle \omega, \phi_j \rangle)\right), \end{array} \right.$$

where $\omega \in \mathcal{F}$ is the parameter we want to estimate.

Several methods for solving this problem without using the mapped image ϕ_j have been proposed. Keerthi *et al.* [8] generate the dual formulation of (2.7) in which the inner products of the mapped images can be replaced by the associated value of the kernel function. They also propose an SMO [11] like algorithm for solving the dual formulation.

On the other hand, Zhu and Hastie [21] first assume that the vector of parameters is given by $\omega = \sum_{j=1}^{l} \alpha_j \phi_j \in \mathcal{F}$. Then, the problem (2.7) can be equivalently written as follows:

$$(2.8) \quad \text{Min.} \quad \frac{\lambda}{2}\alpha^T K\alpha + \sum_{j=1}^{l} \log\left(1 + \exp(-y_j\,(K\alpha)_j)\right),$$

where $(K\alpha)_j$ stands for the j-th element of the vector $K\alpha$. They propose the Newton-Raphson method which approximately minimizes the problem (2.8).

It seems that the dual formulation of (2.7), as well as (2.8), is optimized not so easy as other kernel based problems such as SVM. This is mainly due to the fact that, the kernel matrix K and the optimal solution α^* are, in general, very dense, which makes the problem intractable as the size of the data points gets large. Therefore, most algorithms only work over a small number of selected variables, and generate an approximate solution.

In the next section, exploiting the special structure of the Laplacian matrix of a graph and the associated kernel matrices, we introduce the semi-supervised setting and a very sparse formulation for kernel logistic regression, which can be optimized very efficiently even when the number of data points is large.

3 Semi-Supervised Learning for Kernel Logistic Regression

In this section, let us first introduce the semi-supervised setting. Let us assume that, in addition to the set of l labeled data points, a set of u *unlabeled* data points $x_j \in \mathbf{R}^N$ $(j = l+1, l+2, \ldots, l+u)$ is available. In the sequel, let $M = l+u$ be the total number of the data points, and let us redefine K as an $M \times M$ kernel matrix.

3.1 The Laplacian of a Graph and the Associated Kernel

Let us first introduce a weighted graph G with vertices corresponding to all the data points. When the data point x_i is among the k-nearest-neighbors of x_j or x_j is among those of x_i, we put an edge between the node i and j, and assign a nonnegative weight b_{ij} representing the similarity between the points x_i and x_j, i.e., the larger the weight is, the more similar the two data points are.

The edge weights b_{ij}'s are defined in several ways. For instance, $b_{ij} = 1$ for each edge (i, j) is the simplest way, while we can set $b_{ij} = \exp\left(-\|x_i - x_j\|^2/t^2\right)$, where t is a hyper-parameter. Note that we set $b_{ij} = 0$ if there exists no edge between i and j. Let B be an $M \times M$ symmetric matrix with the elements b_{ij}, and D be a diagonal matrix the i-th diagonal element of which is the sum of the i-th row of the matrix B, i.e.,

$$D_{ii} = \sum_{j=1}^{M} b_{ij}, \quad i = 1, 2, \ldots, M.$$

Then, the Laplacian matrix L of the graph G is defined as

$$L = D - B,$$

which plays a central role for generating graph kernels.

There are several methods for generating kernel matrices based on L. Fouss *et al.* have shown that the pseudoinverse L^+ of L is positive semidefinite, and then L^+ can act as a kernel matrix [3]. They show that, as

long as the graph is connected, L^+ is explicitly given as follows:

$$(3.9)\quad L^+ = \left(L - \frac{ee^T}{M}\right)^{-1} + \frac{ee^T}{M},$$

Also, Zhu *et al.* [24] have introduced the following regularized Laplacian kernel matrix

$$(3.10)\quad K_R = (I + tL)^{-1},$$

where t is a positive parameter.

Moreover, by introducing the modified Laplacian $L_\gamma = \gamma D - B$ with a parameter $0 \leq \gamma \leq 1$, Ito *et al.* [6] defined the modified Laplacian regularized kernel matrix as

$$(3.11)\quad K_{MR} = (I + tL_\gamma)^{-1}.$$

In particular, when $\gamma = 0$ this kernel matrix is the von Neumann diffusion kernel which is defined as

$$(3.12)\quad K_N = \sum_{k=0}^{\infty} t^k B^k = (I - tB)^{-1}.$$

Furthermore, introducing the *normalized Laplacian* $\tilde{L} \equiv D^{-1/2} L D^{-1/2}$, Smola and Kondor [14] propose several kernel matrices such as the diffusion kernel

$$(3.13)\quad K_D = \exp(-\frac{\sigma}{2}\tilde{L})$$

and a normalized variant of the regularized Laplacian kernel defined as follows:

$$(3.14)\quad \tilde{K_R} = \left(I + t\tilde{L}\right)^{-1}.$$

It is rather obvious to see that these kernel matrices are positive semidefinite, which implies that the corresponding mapped images $\phi_j \in \mathcal{F}$ can be calculated explicitly in the following way. In general, let K be one of these kernel matrices with elements $K_{ij} = \langle \phi_i, \phi_j \rangle$. Also, let U be an orthonormal matrix whose columns correspond to the eigenvectors of K, and Λ be a diagonal matrix whose diagonal elements correspond to the eigenvalues of K. Then, one can obtain the eigendecomposition $K = U\Lambda U^T$. This allows us to obtain the mapped image ϕ_j as the j-th column vector of the matrix $\Lambda^{\frac{1}{2}} U^T$. Now, let us define

$$(3.15)\quad P \equiv \Lambda^{\frac{1}{2}} U^T = [\phi_1 \phi_2 \cdots \phi_M].$$

We note that P is an $M \times M$ square matrix and thet the kernel matrix K can be decomposed as $K = P^T P$.

3.2 Semi-Supervised Learning

In this section, let us introduce a semi-supervised approach for kernel logistic regression with the regularized Laplacian kernel (3.10), or its normalized variant (3.14) given in the previous section.

Recall that we have M data points where the first l points are labeled and the rest are unlabeled, and that K is an $M \times M$ positive define matrix defined by (3.10) or (3.14). It should be noted that these matrices are nonsingular. We have also assumed that the mapped data points are denoted by $\phi_j \in \mathcal{F}$ $(j = 1, 2, \ldots, M)$. Then, we can write the minimization problem for logistic regression in the feature space is given as follows:

$$(3.16)\quad \left| \text{Min.} \quad \frac{\lambda}{2}\|\omega\|^2 + \sum_{j=1}^{l} \log\left(1 + \exp\left(-y_j \langle \omega, \phi_j \rangle\right)\right), \right.$$

where $\omega \in \mathcal{F}$. Note that the second term consists of l labeled points.

Next, introducing new l variables $\beta_1, \beta_2, \ldots, \beta_l$, let us define

$$(3.17)\quad \beta_j = \langle \omega\,,\, \phi_j \rangle,\ j = 1, 2, \ldots, l.$$

Furthermore, let us introduce additional u variables $\beta_{l+1}, \beta_{l+2}, \ldots, \beta_M$ and define

$$(3.18)\quad \beta_j = \langle \omega\,,\, \phi_j \rangle,\ j = l+1, l+2, \ldots, M.$$

Now, combining (3.17) and (3.18), we can substitute ω and ϕ_j's in (3.16) with the new variables β_j. It follows from (3.15) that we can write

$$\beta^T = w^T P$$

where $\beta = (\beta_1, \beta_2, \ldots, \beta_M)^T \in \mathbf{R}^M$.

In addition, when K is the regularized Laplacian kernel matrix (3.10) or its normalized variant (3.14), it is obvious to see that K is nonsingular, and so is the matrix P. Multiplying P^{-1}, we have $w^T = \beta^T P^{-1}$, which results in

$$\langle w\,,\, w \rangle = \beta^T P^{-1} (P^{-1})^T \beta = \beta^T K^{-1} \beta.$$

As a result, the problem (3.16) is finally written as follows:

$$(3.19)\quad \text{Min.}\quad \frac{\lambda}{2} \beta K^{-1} \beta + \sum_{j=1}^{l} \log\left(1 + \exp(-y_j \beta_j)\right).$$

Here, it should be emphasized that this formulation for kernel logistic regression has several advantages:

(i) When the kernel matrix is the regularized Laplacian kernel (3.10), or its normalized variant (3.14), the inverse of the kernel matrix K^{-1} is explicitly given by the Laplacian matrix L.

(ii) When the Laplacian matrix L is sparse, so is the objective function of the problem (3.19).

We note that, compared to the conventional formulation (2.8) developed in the *supervised* setting, the proposed formulation (3.19) can be efficiently optimized due to the sparseness of the objective function. In our numerical experiments, we use the scaled conjugate gradient method [1], which works quite well.

Also, let $\beta^* = (\beta_1^*, \beta_2^*, \cdots, \beta_M^*)^T$ be an optimal solution of (3.19). Then,

(iii) the predicted probability of the class label y of the unlabeled data point x_j is simply given as

$$\hat{P}(y|x_j) = \frac{1}{1 + \exp(-y\beta_j^*)},\quad j = l+1, l+2, \ldots, M.$$

Therefore, it is not necessary to obtain the optimal solution ω^* of the problem (3.16).

4 Active Learning

Suppose that we are given a small number of labeled points as well as a pool of *unlabeled* data points. The purpose of active learning is to select a good data point from the pool to assign a label so as to improve the accuracy of the current classifier. In the field of experimental design in statistics, the prediction variance over a pool of unlabeled data points is called the *risk* of the classifier, and is often used as a criterion for selecting a data point. Minimization of this risk is referred to as A-optimality. In this section, we show that active learning based on the A-optimal criterion can be carried out in the feature space defined by the kernel matrices.

4.1 A-optimal Active Learning for Logistic Regression

First, let us describe the A-optimality objective function for the conventional linear logistic regression. Recall that the formulation of linear logistic regression minimizes the regularized negative log likelihood given in (2.3). Let $\hat{w} \in \mathbf{R}^N$ be an estimated parameter, i.e., an optimal solution of (2.3). For each data point x_j in the pool, the associated probability is given by

$$(4.20)\quad \sigma_j(\hat{w}) \equiv P(y=1|x_j) = \frac{1}{1+\exp(-\langle \hat{w}, x_j \rangle)}, \quad j = l+1, \ldots, M.$$

Moreover, for each data point x_j, it is easy to verify that the gradient vector of the associated probability $\sigma_j(\hat{w})$, i.e.,

$$\nabla\sigma_j(\hat{w}) = \left(\frac{\partial\sigma_j(\hat{w})}{\partial w_1}, \frac{\partial\sigma_j(\hat{w})}{\partial w_2}, \ldots, \frac{\partial\sigma_j(\hat{w})}{\partial w_N}\right)^T \in \mathbf{R}^N,$$

is given as follows:

$$\nabla\sigma_j(\hat{w}) = x_j\sigma_j(\hat{w})(1-\sigma_j(\hat{w})).$$

Also, let $F(\hat{w})$ be the observed Fisher information matrix given as:

$$F(\hat{w}) = \sum_{j=1}^{l} x_j x_j^T \sigma_j(\hat{w})(1-\sigma_j(\hat{w})) + \lambda I,$$

where I is the identity matrix. Then, the variance of $\sigma_j(\hat{w})$ is approximately expressed as follows:

$$\mathrm{Var}(\sigma_j(\hat{w})) = \nabla\sigma_j(\hat{w})^T F(\hat{w})^{-1}\nabla\sigma_j(\hat{w}).$$

Therefore, the total risk over the pool is given as follows:

$$(4.21)\quad \mathcal{R}(\hat{w}) = \sum_{j=l+1}^{M} \nabla\sigma_j(\hat{w})^T F(\hat{w})^{-1}\nabla\sigma_j(\hat{w}) = \sum_{j=l+1}^{M} \left(\sigma_j(\hat{w})(1-\sigma_j(\hat{w}))\right)^2 x_j^T F(\hat{w})^{-1} x_j.$$

The purpose of active learning is to select an unlabeled data point in an attempt to minimize the above risk function. Now, let $\hat{w}^{+(x_k,y_k)}$ denote the new estimate which is calculated after adding the data point x_k with the label y_k to the training set. Using the current estimated label probability $\sigma_k(\hat{w})$, the expected risk after adding the data point x_k is given as follows:

$$\hat{\mathcal{R}}(\hat{w}^{+(x_k)}) \equiv \sigma_k(\hat{w})\mathcal{R}(\hat{w}^{+(x_k,1)}) + (1-\sigma_k(\hat{w}))\mathcal{R}(\hat{w}^{+(x_k,0)}).$$

Therfore, in each iteration, the task of active learning is to select the data point x_{k^*} which minimizes the expected risk in the following way:

$$k^* = \mathrm{argmin}\left\{\hat{\mathcal{R}}(\hat{w}^{+(x_k)}) | k = l+1, l+2, \ldots, M\right\}.$$

4.2 Active Learning for Kernel Logistic Regression

Let us now extend the A-optimality criterion for the conventional linear logistic regression to the kernel based logistic regression. In the sequel, let $\hat{\omega}$ be an optimal solution of the problem (3.16) which is defined by the transformed data points $\phi_j \in \mathcal{F}$. Then, the total risk over the pool is given as follows:

$$(4.22)\quad \mathcal{R}(\hat{\omega}) = \sum_{j=l+1}^{M} \left(\sigma_j(\hat{\omega})(1-\sigma_j(\hat{\omega}))\right)^2 \phi_j^T F(\hat{\omega})^{-1}\phi_j,$$

where $F(\hat{\omega})$ is the Fisher information matrix corresponding to the log likelihood given in the problem (3.16).

For simplicity, let us introduce an $M \times M$ diagonal matrix $Z(\hat{\omega}) \in \mathbf{R}^{M \times M}$ each diagonal element of which is given as follows:

$$\{Z(\hat{\omega})\}_{jj} = \begin{cases} \sigma_j(\hat{\omega})(1 - \sigma_j(\hat{\omega})), & j = 1, 2, \ldots, l, \\ 0, & \textit{otherwize.} \end{cases}$$

Now, recall that the $M \times M$ square matrix P defined in (3.15) consists of the mapped data points in its column. Then, the Fisher information matrix is equivalently given by

$$F(\hat{\omega}) = PZ(\hat{\omega})P^T + \lambda I.$$

Now, let us consider the case when the kernel matrix K is nonsingular. Since the kernel matrix is decomposed as $K = P^T P$, we have $I = PK^{-1}P^T$, which results in the following expression for the Fisher information matrix.

$$F(\hat{\omega}) = PZ(\hat{\omega})P^T + \lambda PK^{-1}P^T = P\left(Z(\hat{\omega}) + \lambda K^{-1}\right)P^T.$$

In addition, for simplicity, let us define

$$H(\hat{\omega}) \equiv Z(\hat{\omega}) + \lambda K^{-1},$$

and let $e_j \in \mathbf{R}^M$ be the j-th unit vector. We then have

$$\phi_j^T F(\hat{\omega})^{-1} \phi_j = (Pe_j)^T F(\hat{\omega})^{-1} (Pe_j) = e_j^T H(\hat{\omega})^{-1} e_j,$$

which is the j-th diagonal element of the inverse of $H(\hat{\omega})$. It follows from this equation that the total risk is given as

$$\mathcal{R}(\hat{\omega}) = \sum_{j=l+1}^{M} \left(\sigma_j(\hat{\omega})(1 - \sigma_j(\hat{\omega}))\right)^2 \phi_j^T F(\hat{\omega})^{-1} \phi_j = \sum_{j=l+1}^{M} \left(\sigma_j(\hat{\omega})(1 - \sigma_j(\hat{\omega}))\right)^2 \left(H(\hat{\omega})^{-1}\right)_{jj}. \tag{4.23}$$

Furthermore, when the number of labeled point, l, is small, the Sherman-Morrison-Woodbury identity [5] can be utilized to calculate the inverse of $H(\hat{\omega})$ efficiently. To this end, let $\overline{Z}(\hat{\omega})$ be the $l \times l$ leading principal submatrix of $Z(\hat{\omega})$, and W be an $l \times M$ matrix consisting of the $l \times l$ identity matrix I_l and the $l \times u$ zero matrix $0_{l \times u}$, i.e., $W = [I_l 0_{l \times u}]$. We then have

$$H(\hat{\omega})^{-1} = \left(\lambda K^{-1} + W^T \overline{Z}(\hat{\omega}) W\right)^{-1} = \frac{1}{\lambda}\left(K - KW^T \overline{K}^{-1} WK\right), \tag{4.24}$$

where

$$\overline{K} = \lambda \overline{Z}(\hat{\omega})^{-1} + WKW^T.$$

Therfore, calculating the inverse of the $l \times l$ matrix $\overline{K}$, we can obtain the j-th diagonal element of $H(\hat{\omega})^{-1}$ in the following way:

$$\left(H(\hat{\omega})^{-1}\right)_{jj} = K_{jj} - \overline{k}_j^T \overline{K}^{-1} \overline{k}_j,$$

where

$$\overline{k}_j^T = (K_{1j} K_{2j} \cdots K_{lj}) \in \mathbf{R}^l.$$

We note that, after obtaining the kernel matrix K, the time complexity for calculating the risk $\mathcal{R}(\hat{\omega})$ is $O(l^3 + ul^2)$, while that for the direct calculation for obtaining the inverse of $H(\hat{\omega})$ amounts to $O(M^3)$.

Finally, it should be noted that the diagonal matrix $Z(\hat{\omega})$, as well as the risk $\mathcal{R}(\hat{\omega})$ are defined by the kernel matrix K and the probability $\sigma_j(\hat{\omega})$ which is also given by the optimal solution β^* of the equivalent problem (3.16) as follows:

$$\sigma_j(\hat{w}) = \frac{1}{1 + \exp(-\beta_j^*)}, \quad j = l+1, l+2, \ldots, M.$$

Therefore, we can calculate the risk $\mathcal{R}(\hat{\omega})$ in the feature space without using ϕ_j.

5 Multiclass Classification

In this section, we extend the active learning algorithm for binary classification problems given in the previous section to that for multiclass classification problems, where the labels y_j of the given training data points takes one of three or more values rather than ± 1. We assume that each label y_j takes one of the integer values ranging from 1 to h. Also, let us assume the probability model in the feature space is given as follows:

$$P(y = c|\phi) = \frac{\exp(\langle\, \omega^c\,,\, \phi\,\rangle)}{\sum_{k=1}^{h} \exp(\langle\, \omega^k\,,\, \phi\,\rangle)}, c = 1, 2, \ldots, h,$$

where $\omega^c \in \mathcal{F}$ $(c = 1, 2, \ldots, h)$ are vectors of parameters we need to estimate.

5.1 Multiclass Kernel Logistic Regression

In the same way as the binary case, the parameters are learned by minimizing the sum of the negative log likelihood and the regularized term as follows:

$$(5.25) \quad \text{Min.} \quad \frac{\lambda}{2}\sum_{k=1}^{h}\|\omega^k\|^2 + \sum_{j=1}^{l}\left(-\langle\, \omega^{y_j}\,,\, \phi_j\,\rangle + \log\sum_{k=1}^{h}\exp(\langle\, \omega^k\,,\, \phi_j\,\rangle)\right).$$

Introducing h vectors of variables $\beta^k \in \mathbf{R}^M$ $(k = 1, 2, \ldots, h)$ and letting

$$\beta^{k^T} = \omega^{k^T} P, \quad k = 1, 2, \ldots, h,$$

we can rewrite this problem using the kernel matrix K as follows:

$$(5.26) \quad \text{Min.} \quad \frac{\lambda}{2}\sum_{k=1}^{h}\beta^{k^T} K^{-1}\beta^k + \sum_{j=1}^{l}\left(-\beta_j^{y_j} + \log\sum_{k=1}^{h}\exp\left(\beta_j^k\right)\right).$$

Note that we can optimize this problem efficiently especially when K^{-1} is explicitly given by the Laplacian matrix.

5.2 A-optimality for Multiclass Case

In this section, let us describe the A-optimality risk criterion for the multiclass problems. Suppose that we have obtained optimal parameters $\hat{\omega}^k \in \mathcal{F}$ $(k = 1, 2, \ldots, h)$ of the problem (5.25). Then, for each $j = 1, 2, \ldots, M$ and $c = 1, 2, \ldots, h$, let us denote the estimated probability by

$$\sigma_j^c\left(\hat{\omega}^1, \hat{\omega}^2, \cdots, \hat{\omega}^h\right) \equiv \frac{\exp(\langle\, \hat{\omega}^c\,,\, \phi_j\,\rangle)}{\sum_{k=1}^{h}\exp(\left\langle\, \hat{\omega}^k\,,\, \phi_j\,\right\rangle)}.$$

In the sequel, for notational simplicity, let us denote this probability by $\sigma_j^c(\hat{\omega})$. Moreover, associated with this probability, let us introduce an h-dimensional vector as follows:

$$s_j^c(\omega) \equiv \left(s_{j1}^c(\omega),\ s_{j2}^c(\omega), \cdots, s_{jh}^c(\omega)\right)^T \in \mathbf{R}^h,$$

where

$$s_{jk}^c(\omega) = \begin{cases} -\sigma_j^c(\omega)\sigma_j^k(\omega) & \text{if } k \neq c, \\ \sigma_j^c(\omega)(1 - \sigma_j^k(\omega)) & \text{if } k = c. \end{cases}$$

It is straightforward to verify that the gradient vector of the $\sigma_j^c(\hat{\omega})$ with respect to the vector ω^k is given as follows:

$$\nabla_{\omega^k}\sigma_j^c(\hat{\omega}) = s_{jk}^c(\hat{\omega})\phi_j,$$

which implies that

$$\begin{aligned}\nabla\sigma_j^c(\hat{\omega}) &= \left(\nabla_{\omega^1}\sigma_j^c(\hat{\omega})^T, \nabla_{\omega^2}\sigma_j^c(\hat{\omega})^T, \cdots, \nabla_{\omega^h}\sigma_j^c(\hat{\omega})^T\right)^T \\ &= s_j^c(\hat{\omega}) \otimes \phi_j,\end{aligned}$$

where the symbol $\otimes$ denotes the Kronecker product. As a result, the total risk over the pool is given as follows:

$$\begin{aligned}\mathcal{R}_M(\hat{\omega}) &= \sum_{j=l+1}^{M}\sum_{c=1}^{h}\nabla\sigma_j^c(\hat{\omega})^T F_M(\hat{\omega})^{-1}\nabla\sigma_j^c(\hat{\omega}) \\ &= \sum_{j=l+1}^{M}\sum_{c=1}^{h}(s_j^c(\hat{\omega}) \otimes \phi_j)^T F_M(\hat{\omega})^{-1}(s_j^c(\hat{\omega}) \otimes \phi_j),\end{aligned} \tag{5.27}$$

where $F_M(\hat{\omega})$ is the Fisher information matrix associated with the problem (5.25). For each $i, k = 1, 2, \ldots, h$, let us define

$$F^{ik}(\omega) \equiv \begin{cases} \sum_{j=1}^{l}\phi_j\phi_j^T\sigma_j^i(\omega)\sigma_j^k(\omega) & \text{if } i \neq k, \\ \sum_{j=1}^{l}\phi_j\phi_j^T\sigma_j^i(\omega)(1-\sigma_j^k(\omega)) + \lambda I & \text{if } i = k. \end{cases} \tag{5.28}$$

Then, the Fisher information matrix for the multiclass problem is given as follows:

$$F_M(\omega) \equiv \begin{bmatrix} F^{11}(\omega) & F^{12}(\omega) & \cdots & F^{1h}(\omega) \\ F^{21}(\omega) & F^{22}(\omega) & \cdots & F^{2h}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ F^{h1}(\omega) & F^{h2}(\omega) & \cdots & F^{hh}(\omega) \end{bmatrix}.$$

In the same way as we have demonstrated in the previous section, the risk given in (5.27) can be calculated only by the kernel matrix without using the feature vector ϕ_j. To this end, for each $c, k = 1, 2, \ldots, h$, let us first introduce an $M \times M$ diagonal matrix $Z^{ck}(\omega)$ the j-th diagonal element of which is defined as

$$\left\{Z^{ck}(\omega)\right\}_{jj} = \begin{cases} \sigma_j^c(\omega)\sigma_j^k(\omega) & \text{if } k \neq c, \\ \sigma_j^c(\omega)(1-\sigma_j^k(\omega)) & \text{if } k = c, \end{cases}$$

for all $j = 1, 2, \ldots, l$, and $\left\{Z^{ck}(\omega)\right\}_{jj} = 0$, for all $j = l+1, l+2, \ldots, M$. Also, let us define the following block matrix with size $hM \times hM$

$$Z_M(\omega) \equiv \begin{bmatrix} Z^{11}(\omega) & Z^{12}(\omega) & \cdots & Z^{1h}(\omega) \\ Z^{21}(\omega) & Z^{22}(\omega) & \cdots & Z^{2h}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ Z^{h1}(\omega) & Z^{h2}(\omega) & \cdots & Z^{hh}(\omega) \end{bmatrix}.$$

It follows that the Fisher information matrix for multiclass problem is equivalently given as

$$F_M(\omega) = (I_h \otimes P)\, H_M(\omega)\, (I_h \otimes P)^T,$$

where

$$H_M(\omega) = \lambda\left(I_h \otimes K^{-1}\right) + Z_M(\omega)$$

and I_h is the identity matrix of size h. Substituting this into (5.27), the risk is finally given as

$$\mathcal{R}_M(\hat{\omega}) = \sum_{j=l+1}^{M}\sum_{c=1}^{h}(s_j^c(\hat{\omega}) \otimes e_j)^T H_M(\hat{\omega})^{-1}(s_j^c(\hat{\omega}) \otimes e_j). \tag{5.29}$$

Here, it should be mentioned that the vectors $s_j^c(\hat{\omega})$ as well as the matrix $H_M(\hat{\omega})^{-1}$ appeared in the above expression are given by the kernel matrix K and the estimated probabilities $\sigma_j^c(\hat{\omega})$ which is also given by the solutions $\beta^{1*}, \beta^{2*}, \ldots, \beta^{h*}$ of the associated problem (5.26) as follows:

$$\sigma_j^c(\hat{\omega}) = \frac{\exp(\beta^{c*}_j)}{\sum_{k=1}^h \exp(\beta^{k*}_j)}.$$

Moreover, it should be noted that the Sherman-Morrison-Woodbury identity used in (4.24) can be further generalized to calculate the inverse of the $Mh \times Mh$ matrix $H_M(\hat{\omega})$ by the $lh \times lh$ matrix in the following way. First, for each $c, k = 1, 2, \ldots, h$, let $\tilde{Z}^{ck}(\omega)$ be the $l \times l$ leading principal submatrix of $Z^{ck}(\omega)$. Also, let us define the block matrix with size $lh \times lh$ as follows:

$$\overline{Z}_M(\omega) \equiv \begin{bmatrix} \tilde{Z}^{11}(\omega) & \tilde{Z}^{12}(\omega) & \cdots & \tilde{Z}^{1h}(\omega) \\ \tilde{Z}^{21}(\omega) & \tilde{Z}^{22}(\omega) & \cdots & \tilde{Z}^{2h}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{Z}^{h1}(\omega) & \tilde{Z}^{h2}(\omega) & \cdots & \tilde{Z}^{hh}(\omega) \end{bmatrix}.$$

Then, we have the following equation:

$$\begin{aligned} H_M(\hat{\omega})^{-1} &= \left(\lambda \left(I_h \otimes K^{-1}\right) + Z_M(\omega)\right)^{-1} \\ &= \frac{1}{\lambda}\left(I_h \otimes K^{-1} + \frac{1}{\lambda}\left(I_h \otimes W^T\right)\overline{Z}_M(\omega)\left(I_h \otimes W^T\right)^T\right)^{-1} \\ &= \frac{1}{\lambda}\left(I_h \otimes K - \left(I_h \otimes KW^T\right)\overline{K}_M(\omega)^{-1}\left(I_h \otimes WK\right)\right) \end{aligned}$$

where

$$\overline{K}_M(\omega) \equiv \lambda \overline{Z}_M(\omega) + I_h \otimes WKW^T.$$

As a result, the risk is given as follows:

$$\mathcal{R}_M(\hat{\omega}) = \frac{1}{\lambda}\sum_{j=l+1}^{M}\sum_{c=1}^{h} s_j^c(\hat{\omega})^T s_j^c(\hat{\omega}) K_{jj} - \left(k_j^T \otimes s_j^c(\hat{\omega})\right)^T \overline{K}(\omega)^{-1}\left(k_j^T \otimes s_j^c(\hat{\omega})\right)$$

Now, let $\hat{w}^{+(x_k,c)}$ denote the estimate calculated after adding the data point x_k with the label c to the training set. Using the current estimated class probability $\sigma_k^c(\hat{w})$, the expected risk after adding the data point x_k is given as follows:

$$\hat{\mathcal{R}}_M(\hat{w}^{+(x_k)}) \equiv \sum_{c=1}^{h} \sigma_k^c(\hat{w}) \hat{\mathcal{R}}_M(\hat{w}^{+(x_k,c)}).$$

Therfore, in each iteration, the task of active learning is to select the data point x_{k^*} which minimizes the expected risk in the following way:

$$(5.30)\quad k^* = \operatorname{argmin}\left\{\hat{\mathcal{R}}_M(\hat{w}^{+(x_k)}) | k = l+1, l+2, \ldots, M\right\}.$$

6 Numerical Experiments

In this section, we will report the performance of the proposed active learning method for multiclass classification. The experiments are conducted on four real-world datasets: Forest Cover Type (FOREST), UseNet news articles (20Newsgroups), USPS hand written digits (USPS), and Splice Junction Gene Sequence (SJGS). These are commonly used benchmark datasets for classification.

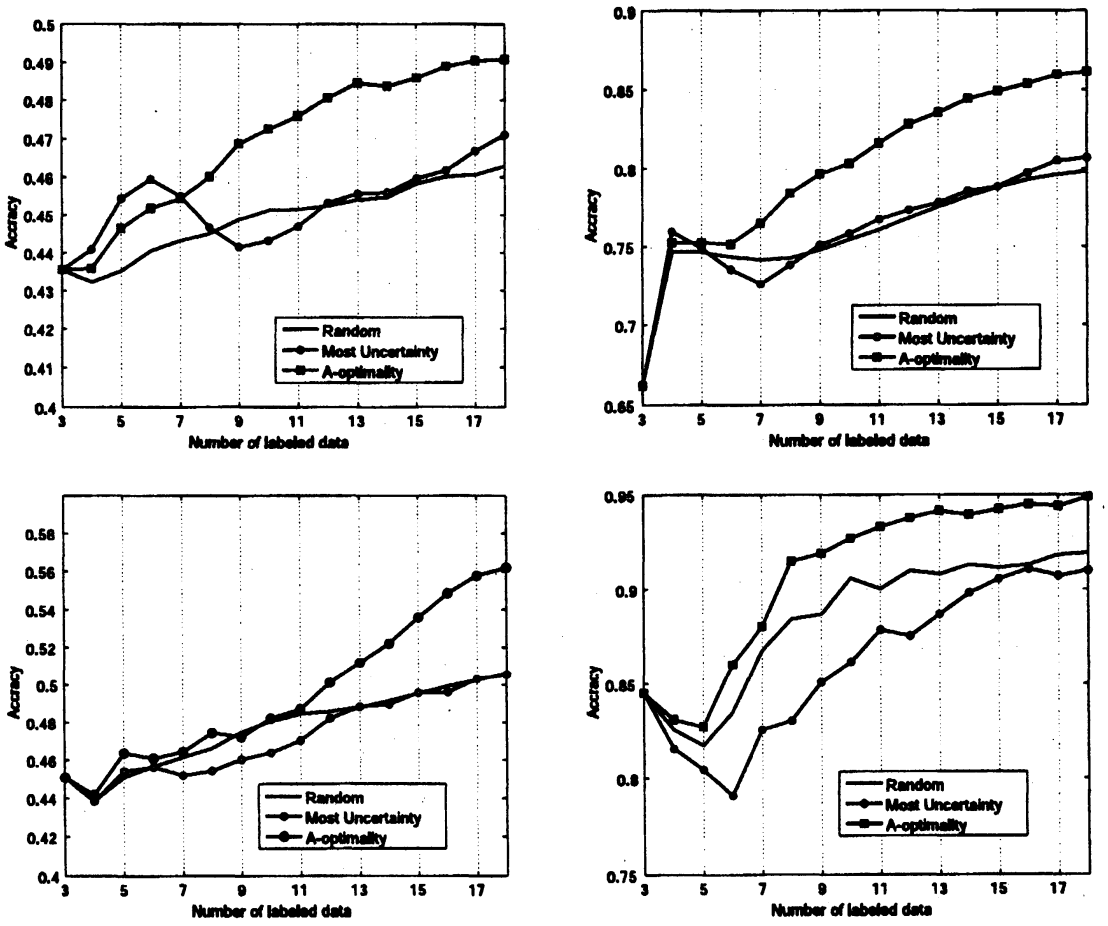

Figure 1: Average accuracies. Top left: FOREST. Top right: 20Newsgroups. Bottom left: SJGS. Bottom right: USPS

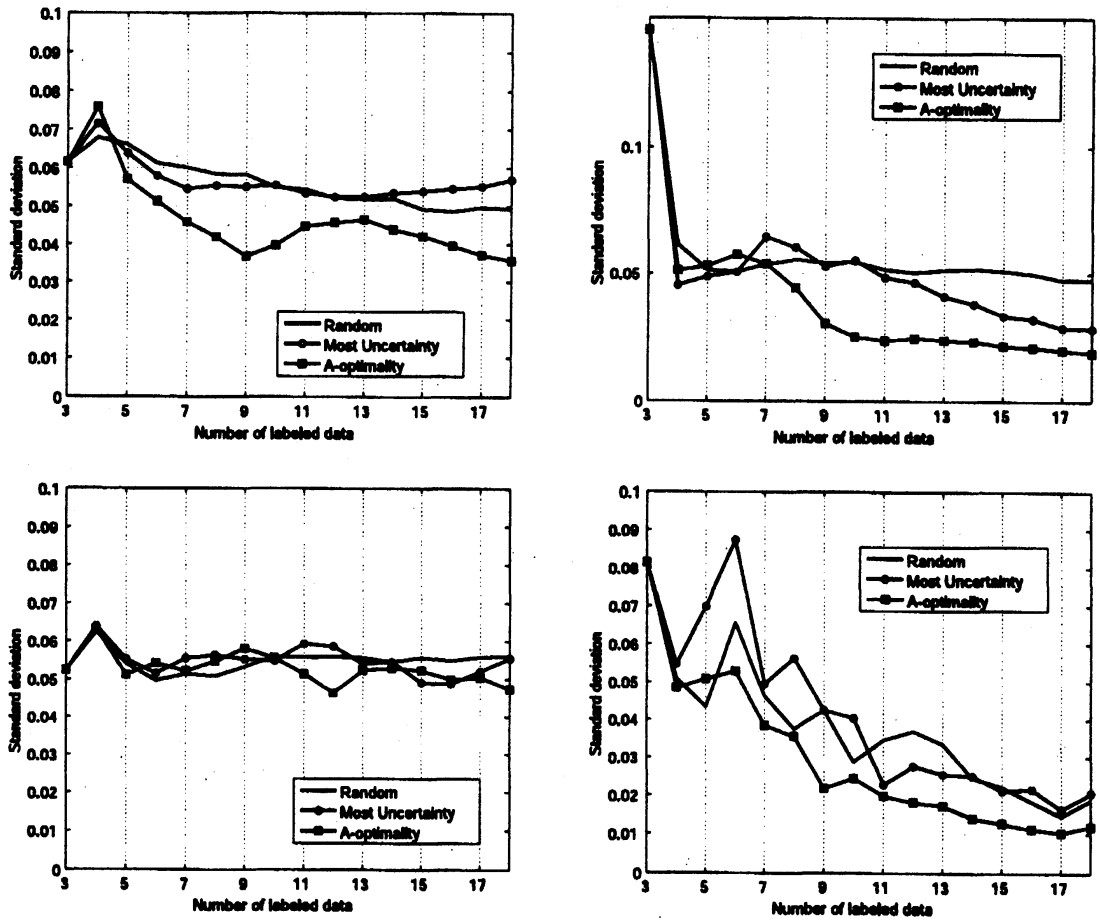

Figure 2: Standard deviations. Top left: FOREST. Top right: 20Newsgroups. Bottom left: SJGS. Bottom right: USPS

The FOREST dataset contains 581,012 instances of the forest cover type for 30 × 30 meter cells, each of which is represented by 10 quantitative and 44 binary variables, and is categorized into seven possible classes. We use the instances of the three largest classes, Spruce-Fir, Lodgepole Pine, and Ponderosa Pine.

The 20Newsgroups dataset has about 20,000 articles which is divided almost evenly among 20 newsgroups. We delete articles belonging to more than two newsgroups, and generate dataset consisting of 18,597 articles. After stemming and stop word removal, we use all terms that occur in more than three articles. We represent each article as a vector of the terms with the TF-IDF term weighting. We use the articles in three newsgroups, `rec.motorcycles`, `rec.sport.baseball` and `rec.sport.hockey`.

The USPS dataset contains 9,298 digit images of size 16 × 16 pixels, which are converted to vectors of size 256. We use digits 1, 2, and 3 in the experiments.

The SJGS dataset consists of 3,190 sequences of DNA. Each sequence has 60 symbolic characters representing the nucleotides of eight types. Each character is replaced by 8 binary indicator variables. Thus, the sequences are given as the binary vectors of size 480. We use all the three classes, EI, IE, and no boundary.

For each dataset, selecting 100 instances randomly in each class, we construct three-class classification problem with 300 instances. We then build a k-nearest-neighbor graph with $k = 6$ using the Euclidian distances for the FOREST, USPS and SJGS datasets, while the cosine similarity is used for the 20Newsgroups dataset. For all datasets, we use the simple weight $b_{ij} = 1$ for generating the matrix B. All the experiments are carried out using the kernel matrix given in (3.14) with $t = 1.0$.

Each active learning trial starts with a set of the randomly selected three initial labeled data points, i.e, one data point for each class. Then, optimizing the problem (5.26), we obtain the estimates of the class probabilities for all data points. An index of the query data point is given by (5.30). After selecting a new query, we add it to the training set with the true label and re-estimate the class probabilities. We iterate this procedure until 15 queries are selected. In this experiments, We perform 10 active learning trials by changing the initial labeled data points. We show the average accuracies over the pool as well as the standard deviations in each iteration.

We compare the proposed method with two baselines: the random query (RQ) and the most uncertain query (MUQ) strategies. In RQ, we select the query randomly from the pool, while, in MUQ, we select the query $\boldsymbol{x}_j$ with the highest entropy, i.e., $-\sum_{c=1}^{3} \hat{P}(c|\boldsymbol{x}_j) \log \hat{P}(c|\boldsymbol{x}_j)$, where $\hat{P}(c|\boldsymbol{x}_j)$ is the estimated probability of the class label c.

The average accuracies and the standard deviations of three methods are shown in Figs. 1 and 2, respectively. We can see from these figures that the proposed active learning method outperforms the two baselines for all datasets in these experiments. It is obvious to see that the proposed active learning method leads to apparently higher accuracies than RQ or MUQ after adding only 9 queries i.e., 12 labeled data points. It seems that the proposed active learning method has an ability to improve the accuracies of the classifiers within a very small number of queries. Also, surprisingly, the accuracies of MUQ is almost the same as that of RQ for three datasets, and even worse than RQ for the USPS datasets. Moreover, we can see that, except for the SJGS datasets, the standard deviations of the proposed active learning method decrease to almost the half of the two baselines after 15 queries are generated. This also indicates the robustness of the proposed active learning method.

7 Conclusion

We have proposed new formulations for kernel logistic regression in the semi-supervised setting, as well as methods for performing pool-based active learning based on the A-optimality criterion for both binary and multiclass classification problems. Exploiting the special structure of the Laplacian kernel matrices, we show that the minimization of the regularized negative log likelihood can be formulated as the very sparse optimization problems. Also, we have demonstrated that A-optimality based active learning can be carried out in the feature

space characterized by the associated kernel matrices. Numerical experiments on the four multiclass classification datasets indicate that the proposed active learning method generates more accurate and robust classifiers than those generated by the RQ or MUQ strategies.

References

[1] D. BERTSEKAS, ed., *Nonlinear Programming*, Athena Scientific, 1999.

[2] D. A. COHN, *Neural network exploration using optimal experiment design*, in Advances in Neural Information Processing Systems, J. D. Cowan, G. Tesauro, and J. Alspector, eds., vol. 6, Morgan Kaufmann Publishers, Inc., 1994, pp. 679–686.

[3] F. FOUSS, A. PIROTTE, AND M. SAERENS, *A novel way of computing dissimilarities between nodes of a graph, with application to collaborative filtering*, in 15th European Conference on Machine Learning (ECML 2004); Proceedings of the Workshop on Statistical Approaches for Web Mining (SAWM), 2004, pp. 26–37.

[4] A. E. HOERL AND R. W. KENNARD, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics, 12 (1970), pp. 55–67.

[5] R. A. HORN AND C. R. JOHNSON, *Matrix analysis*, Cambridge University Press, Cambridge-New York, 1985.

[6] T. ITO, M. SHIMBO, T. KUDO, AND Y. MATSUMOTO, *Application of kernels to link analysis*, in KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, New York, NY, USA, 2005, ACM Press, pp. 586–592.

[7] T. JOACHIMS, *Making large-scale support vector machine learning practical*, in Advances in Kernel Methods, B. Schölkopf, C. Burges, and A. Smola, eds., The MIT Press, 1999, pp. 169–184.

[8] S. S. KEERTHI, K. DUAN, S. K. SHEVADE, AND A. N. POO, *A fast dual algorithm for kernel logistic regression*, in Proceedings of the 19th International Conference on Machine Learning, 2002, pp. 299–306.

[9] D. D. LEWIS AND W. A. GALE, *A sequential algorithm for training text classifiers*, in SIGIR, 1994, pp. 3–12.

[10] D. MACKAY, *Information-based objective functions for active data selection*, Neural Computation, 4 (1992), pp. 590–604.

[11] J. C. PLATT, *Fast training of support vector machines using sequential minimal optimization*, in Advances in Kernel Methods, B. Schölkopf, C. Burges, and A. Smola, eds., The MIT Press, 1999, pp. 185–208.

[12] H. S. SEUNG, M. OPPER, AND H. SOMPOLINSKY, *Query by committee*, in COLT, 1992, pp. 287–294.

[13] J. SHAWE-TAYLOR AND N. CRISTIANINI, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, 2004.

[14] A. SMOLA AND I. KONDOR, *Kernels and regularization on graphs*, in Proceedings of the Annual Conference on Computational Learning Theory, B. Schölkopf and M. Warmuth, eds., Lecture Notes in Computer Science, Springer, 2003.

[15] S. TONG AND D. KOLLER, *Support vector machine active learning with applications to text classification*, Journal of Machine Learning Research, 2 (2001), pp. 45–66.

[16] V. N. VAPNIK, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[17] Y. YAJIMA, *One-class support vector machines for recommendation tasks*, in 10th Pacific-Asia Conference, PAKDD, 2006, pp. 230–239.

[18] Y. YAJIMA AND T. HOSHIBA, *Optimization approaches for semi-supervised learning*, in ICMLA, 2005.

[19] T. ZHANG AND F. OLES, *Text categorization based on regularized linear classification methods*, Information Retrieval, 4 (2004), pp. 5–31.

[20] D. ZHOU, O. BOUSQUET, T. N. LAL, J. WESTON, AND B. SCHÖLKOPF, *Learning with local and global consistency*, Advances in Neural Information Processing Systems, 16 (2004), pp. 321–328.

[21] J. ZHU AND T. HASTIE, *Kernel logistic regression and the import vector machine.*, in NIPS, 2001, pp. 1081–1088.

[22] X. ZHU, Z. GHAHRAMANI, AND J. LAFFERTY, *Semi-supervised learning using Gaussian fields and harmonic functions.*, in ICML, 2003, pp. 912–919.

[23] X. ZHU, J. LAFFERTY, AND Z. GHAHRAMANI, *Combining active learning and semi-supervised learning using gaussian fields and harmonic functions*, in ICML-2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, 2003, pp. 58–65.

[24] ——, *Semi-supervised learning: From Gaussian fields to Gaussian processes*, Technical Report CMU-CS-03-175, Carnegie Mellon University, 2003.