The Korean Communications in Statistics **Vol. 14** No. 3, 2007, pp. 507–516

Multinomial Kernel Logistic Regression via Bound Optimization Approach*

Jooyong Shim[1], Dug Hun Hong[2], Dal Ho Kim[3] and Changha Hwang[4]

Abstract

Multinomial logistic regression is probably the most popular representative of probabilistic discriminative classifiers for multiclass classification problems. In this paper, a kernel variant of multinomial logistic regression is proposed by combining a Newton's method with a bound optimization approach. This formulation allows us to apply highly efficient approximation methods that effectively overcomes conceptual and numerical problems of standard multiclass kernel classifiers. We also provide the approximate cross validation (ACV) method for choosing the hyperparameters which affect the performance of the proposed approach. Experimental results are then presented to indicate the performance of the proposed procedure.

Keywords: Approximate cross validation; hyperparameters; multinomial logistic regression; support vector machine.

1. Introduction

Classifiers can be partitioned into two main groups, namely informative and discriminative ones. Classical linear discriminant analysis (LDA) is the most popular informative method, whereas logistic regression is the most popular discriminative one. In general, logistic regression is more robust than LDA, since less assumptions about the classes are made. An important advantage of logistic

* This work was supported by grant R01-2006-000-10226-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

[1] Adjunct Professor, Department of Applied Statistics, Catholic University of Daegu, Kyungbuk 712-702, Korea.

[2] Professor, Department of Mathematics, Myongji University, Kyunggido 449-72, Korea.

[3] Professor, Department of Statistics, Kyungbuk National University, Daegu 702-701, Korea.

[4] Professor, Division of Information and Computer Science, Dankook University, Seoul 140-714, Korea.

Correspondence : chwang@dankook.ac.kr

regression is that it outputs an estimate of the probability that an object belongs to each of the possible classes. A different approach to discriminative classification is the support vector machine (SVM) by Vapnik (1995, 1998). Compared to logistic regression, the main drawback of the SVM is the absence of probabilistic outputs. The most popular methods for multiclass classification in recent machine learning research are variants on SVMs and boosting, sometimes combined with error-correcting code approaches. Rifkin and Klautau (2004) provide a review.

In this paper we focus on multinomial generalization of logistic regression and on a nonlinear "kernelized" variant of multinomial logistic regression. Using a bound optimization approach as in Krishnapuram *et al.* (2005), we derive a fast exact algorithm for learning multinomial kernel logistic regression (MKLR). Concerning multiclass problems, the availability of probabilistic outputs allows us to overcome the main drawback of the SVM: in the usual SVM framework, a multiclass problem with m classes is treated as a series of binary classification methods such as the one-vs-one and one-vs-all, or the single machine type methods which attempt to construct a multiclass classifier by solving a single optimization problem. There is no substantial agreement on which method is the best one for the multiclass problem. See for details Weston and Watkins (1998), Suykens and Vandewalle (1999) and Rifkin and Klautau (2004). We also present the approximate cross validation (ACV) method for choosing the hyperparameters which affect the performance of the proposed MKLR.

2. Multinomial Logistic Regression

Let $\boldsymbol{x} = (x_1, \ldots, x_d)^T$ be an input vector to be classified. We encode the fact that an input vector belongs to a class $k \in \{1, \ldots, m\}$ by a m-dimensional 0/1 valued vector $\boldsymbol{y} = (y_1, \ldots, y_m)^T$, where $y_k = 1$ and all other coordinates are 0. Multinomial logistic regression is a conditional probability model of the form

$$P(y_k = 1 | \boldsymbol{x}, \boldsymbol{\omega}) = \frac{\exp(\boldsymbol{\omega}_k^T \boldsymbol{x})}{\sum_{j=1}^{m} \exp(\boldsymbol{\omega}_j^T \boldsymbol{x})}, \tag{2.1}$$

parameterized by the dm-dimensional vector $\boldsymbol{\omega} = (\boldsymbol{\omega}_1^T, \ldots, \boldsymbol{\omega}_m^T)^T$, where $\boldsymbol{\omega}_k$ is the d-dimensional weight vector corresponding to class k and the superscript ^{T} denotes vector or matrix transpose. This is a direct generalization of binary logistic regression to the multiclass case. Since the probabilities must sum to one: $\sum_{k=1}^{m} P(y_k = 1|\boldsymbol{x}, \boldsymbol{\omega}) = 1$, the weight vector for one of the classes need not be estimated. Without loss of generality, we thus set $\boldsymbol{\omega}_m = \mathbf{0}$ and the only parameters to be learned are the weight vectors $\boldsymbol{\omega}_k$ for $k \in \{1, \ldots, m-1\}$. For the remainder of the paper, we use $\boldsymbol{\omega}$ to denote the $d(m-1)$-dimensional vector of parameters to be learned.

Classification of a new observation is based on the vector of conditional probability estimates produced by the model. In this paper we simply assign the class with the highest conditional probability estimate:

$$\hat{y}(\boldsymbol{x}) = \arg\max_k P(y_k = 1|\boldsymbol{x}). \tag{2.2}$$

Consider a set of training examples $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d, \boldsymbol{y}_i \in \mathbb{R}^m$. Maximum likelihood estimation of the parameters $\boldsymbol{\omega}$ is equivalent to minimizing the negative log-likelihood function:

$$\ell(\boldsymbol{\omega}) = -\sum_{i=1}^{n}\sum_{k=1}^{m-1} y_{ik}\boldsymbol{\omega}_k^T\boldsymbol{x}_i + \sum_{i=1}^{n}\log\left(1 + \sum_{k=1}^{m-1}\exp(\boldsymbol{\omega}_k^T\boldsymbol{x}_i)\right), \tag{2.3}$$

which is typically accomplished using Newton's method, also known, in this case, as iteratively reweighted least squares (IRWLS). Although there are other methods for performing this minimization, none clearly outperforms IRWLS (Minka, 2003). See Böhning (1992) and Krishnapuram *et al.* (2005) for details of estimating $\boldsymbol{\omega}$ in multinomial logistic regression.

3. Multinomial Kernel Logistic Regression

A nonlinear form of multinomial logistic regression, known as multinomial kernel logistic regression, can be obtained via the so-called "kernel trick", whereby a conventional multinomial logistic regression model is constructed in a high dimensional feature space induced by a Mercer (1909)'s kernel. More formally, given training data, $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d, \boldsymbol{y}_i \in \mathbb{R}^m$, a feature space $\mathcal{F}$ $(\boldsymbol{\phi} : \mathcal{X} \to \mathcal{F})$, is defined by a kernel function, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, that evaluates the inner product between the images of input vectors in the feature space, *i.e.* $K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \boldsymbol{\phi}(\boldsymbol{x}_k)^T\boldsymbol{\phi}(\boldsymbol{x}_l)$. The kernel function used here is the Gaussian kernel,

$$K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \exp\left(-\frac{1}{\sigma^2}\|\boldsymbol{x}_k - \boldsymbol{x}_l\|^2\right),$$

where σ^2 is the kernel parameter.

The negative log-likelihood function of the multinomial logistic regression model constructed in the feature space is given as follows:

$$\ell(\boldsymbol{\eta}) = -\sum_{i=1}^{n}\sum_{k=1}^{m-1} y_{ik}\eta_{ik} + \sum_{i=1}^{n}\log\left(1+\sum_{k=1}^{m-1}\exp(\eta_{ik})\right), \ \eta_{ik} = \boldsymbol{\omega}_k^T\boldsymbol{\phi}(\boldsymbol{x}_i) \quad (3.1)$$

Often, model penalization improves generalization performance and so we employ the following penalty to the negative log-likelihood function during model fitting:

$$\ell(\boldsymbol{\eta},\boldsymbol{\omega}) = -\sum_{i=1}^{n}\sum_{k=1}^{m-1} y_{ik}\eta_{ik} + \sum_{i=1}^{n}\log\left(1+\sum_{k=1}^{m-1}\exp(\eta_{ik})\right) + \frac{\lambda}{2}\sum_{k=1}^{m-1}\|\boldsymbol{\omega}_k\|^2, \quad (3.2)$$

where λ is a penalty parameter which controls the trade-off between the goodness-of-fit on the data and the smoothness.

The representation theorem (Kimeldorf and Wahba, 1971) guarantees that the minimizer of the penalized negative log-likelihood (3.2) to be $\eta_{ik} = \boldsymbol{K}_i^T\boldsymbol{\alpha}_k$, where $\boldsymbol{K}_i$ is the ith column of the kernel matrix $\boldsymbol{K}$ with elements $K(\boldsymbol{x}_k, \boldsymbol{x}_l)$. Now the problem becomes obtaining the $(m-1)n$-dimensional vector $\boldsymbol{\alpha}$ to minimize

$$\begin{aligned}\ell(\boldsymbol{\alpha}) = & -\sum_{i=1}^{n}\sum_{k=1}^{m-1} y_{ik}\boldsymbol{K}_i^T\boldsymbol{\alpha}_k + \sum_{i=1}^{n}\log\left(1+\sum_{k=1}^{m-1}\exp\left(\boldsymbol{K}_i^T\boldsymbol{\alpha}_k\right)\right) \\ & +\frac{\lambda}{2}\sum_{k=1}^{m-1}\boldsymbol{\alpha}_k^T\boldsymbol{K}\boldsymbol{\alpha}_k, \end{aligned} \quad (3.3)$$

where $\boldsymbol{\alpha}$ is denoted as $(\boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_{m-1}^T)^T$.

Let $p_{ij} = \exp\left(\boldsymbol{K}_i^T\boldsymbol{\alpha}_j\right)/(1+\sum_{l=1}^{m-1}\exp\left(\boldsymbol{K}_i^T\boldsymbol{\alpha}_l\right))$ and then let us define $\boldsymbol{p}_{i\cdot} = (p_{i1}, \ldots, p_{i,m-1})^T$ and $\boldsymbol{p}_{\cdot k} = (p_{1k}, \ldots, p_{nk})^T$. Then, using Newton's method, the optimal $\boldsymbol{\alpha}$ can be obtained iteratively as follows:

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - (\boldsymbol{H}^* + \frac{1}{\lambda}\boldsymbol{K}^*)^{-1}\boldsymbol{G}, \quad (3.4)$$

where $\boldsymbol{K}^* = \text{diag}(\boldsymbol{K}, \ldots, \boldsymbol{K})$, and $\boldsymbol{H}^*$ and $\boldsymbol{G}$ are defined as

$$\boldsymbol{H}^* = \boldsymbol{K}^*\left(\text{diag}\begin{pmatrix}\boldsymbol{p}_{\cdot 1}^{(t)}\\ \vdots \\ \boldsymbol{p}_{\cdot m-1}^{(t)}\end{pmatrix} - \begin{bmatrix}\text{diag}(\boldsymbol{p}_{\cdot 1}^{(t)})\\ \vdots \\ \text{diag}(\boldsymbol{p}_{\cdot m-1}^{(t)})\end{bmatrix}\left[\text{diag}(\boldsymbol{p}_{\cdot 1}^{(t)}), \ldots, \text{diag}(\boldsymbol{p}_{\cdot m-1}^{(t)})\right]\right)\boldsymbol{K}^*,$$

$$\boldsymbol{G} = \boldsymbol{K}^*\left(-\begin{pmatrix}\boldsymbol{y}_{\cdot 1}^{(t)}\\ \vdots \\ \boldsymbol{y}_{\cdot m-1}^{(t)}\end{pmatrix} + \begin{pmatrix}\boldsymbol{p}_{\cdot 1}^{(t)}\\ \vdots \\ \boldsymbol{p}_{\cdot m-1}^{(t)}\end{pmatrix} + \frac{1}{\lambda}\begin{pmatrix}\boldsymbol{\alpha}_{1}^{(t)}\\ \vdots \\ \boldsymbol{\alpha}_{m-1}^{(t)}\end{pmatrix}\right).$$

Here $\boldsymbol{y}_{\cdot k}$ is defined as $\boldsymbol{y}_{\cdot k} = (y_{1k}, \dots, y_{nk})^T$. It is noted that $\boldsymbol{H}^*$ can be rewritten as

$$\boldsymbol{H}^* = \sum_{i=1}^{n} (\mathrm{diag}(\boldsymbol{p}_{i\cdot}^{(t)}) - \boldsymbol{p}_{i\cdot}^{(t)} \boldsymbol{p}_{i\cdot}^{(t)T}) \otimes \boldsymbol{K}_i \boldsymbol{K}_i^T, \tag{3.5}$$

where *otimes* is the Kronecker matrix product.

As shown in Böhning (1992) and Krishnapuram *et al.* (2005), the Hessian of the negative log-likelihood is upper bounded by a positive definite matrix that does not depend on $\boldsymbol{\alpha}$,

$$\boldsymbol{H}^* \leq \frac{1}{2}(\boldsymbol{I} - \mathbf{1}\mathbf{1}^T/m) \otimes \sum_{i=1}^{n} \boldsymbol{K}_i \boldsymbol{K}_i^T \equiv \boldsymbol{B}^*.$$

Here, $\boldsymbol{A} \leq \boldsymbol{B}$ means $\boldsymbol{A} - \boldsymbol{B}$ is negative semidefinite. Thus, using the bound optimization technique, we have a simple Newton's method for $\boldsymbol{\alpha}$,

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - (\boldsymbol{B}^* + \frac{1}{\lambda}\boldsymbol{K}^*)^{-1}\boldsymbol{G}. \tag{3.6}$$

Using upper bound matrix $\boldsymbol{B}^*$, we do not need to compute Hessian at each iteration, which yields a fast algorithm for the multinomial kernel logistic regression.

4. Approximate Cross Validation

Model selection, the process of determining the optimal regularization and kernel parameters, is a central issue in fitting kernel machines. The goal of model selection is to identify the model that will yield the best generalization performance. In this section, we present a novel approximation to the leave-one-out (LOO) error estimator that is an important statistical tool for assessing generalization performance of the multinomial kernel logistic regression.

Define the cross validation (CV) function as

$$\mathrm{CV}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \sum_{k=1}^{m-1} y_{ik} \eta_{ik}^{(-i)} + \sum_{i=1}^{n} b(\boldsymbol{\eta}_{i\cdot}),$$

where $\boldsymbol{\theta}$ is a set of hyperparameters and $b(\boldsymbol{\eta}_{i\cdot}) = \log(1 + \sum_{k=1}^{m-1} \exp(\eta_{ik}))$ with $\boldsymbol{\eta}_{i\cdot} = (\eta_{i1}, \dots, \eta_{i,m-1})^T$. The CV function can be rewritten as

$$\mathrm{CV}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \sum_{k=1}^{m-1} y_{ik} \eta_{ik} + \sum_{i=1}^{n} b(\boldsymbol{\eta}_{i\cdot}) + \sum_{i=1}^{n} \boldsymbol{y}_{i\cdot}^T (\boldsymbol{\eta}_{i\cdot} - \boldsymbol{\eta}_{i\cdot}^{(-i)}),$$

where $\boldsymbol{\eta}_{i\cdot}^{(-i)} = (\eta_{i1}^{(-i)}, \eta_{i2}^{(-i)}, \ldots, \eta_{i,m-1}^{(-i)})^T$, $\eta_{ik}^{(-i)} = \boldsymbol{K}_i^{(-i)T}\boldsymbol{\alpha}_k^{(-i)}$, $\boldsymbol{K}_i^{(-i)} = (K(\boldsymbol{x}_1, \boldsymbol{x}_i)$ $, \ldots, K(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i), K(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i), \ldots, K(\boldsymbol{x}_n, \boldsymbol{x}_i))^T$ and $\boldsymbol{\alpha}_k^{(-i)}$ is obtained by minimizing $L(\boldsymbol{\alpha})$ without the ith observation.

Let $\tilde{\boldsymbol{\eta}} = (\boldsymbol{\eta}_{\cdot 1}^T, \ldots, \boldsymbol{\eta}_{\cdot m-1}^T)^T$ and $\tilde{\boldsymbol{y}} = (\boldsymbol{y}_{\cdot 1}^T, \ldots, \boldsymbol{y}_{\cdot m-1}^T)^T$ where $\boldsymbol{\eta}_{\cdot k} = (\eta_{1k}, \ldots, \eta_{nk})^T$ and $\boldsymbol{y}_{\cdot k} = (y_{1k}, \ldots, y_{nk})^T$, and let $\tilde{\boldsymbol{y}}^{(-i)}$ be the $n(m-1) \times 1$ vector consisting of $\boldsymbol{y}_{\cdot k}$ replaced the ith element with $p_{ik}^{(-i)}$. For example, $\tilde{\boldsymbol{y}}^{(-1)} = (p_{11}^{(-1)}, y_{21}, \ldots, y_{n1}, p_{12}^{(-1)}, y_{22}, \ldots, y_{n2}, \ldots, p_{1,m-1}^{(-1)}, y_{2,m-1}, \ldots, y_{n,m-1})^T$. Let $\tilde{\boldsymbol{\eta}}^{(-i)} = (\boldsymbol{\eta}_{\cdot 1}^{(-i)T}, \boldsymbol{\eta}_{\cdot 2}^{(-i)T}, \ldots, \boldsymbol{\eta}_{\cdot m-1}^{(-i)T})^T$ where $\boldsymbol{\eta}_{\cdot k}^{(-i)} = (\eta_{1k}^{(-i)}, \ldots, \eta_{nk}^{(-i)})^T$. Then the penalized negative log-likelihood can be rewritten as

$$L(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\eta}}) = -\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{\eta}} + b(\tilde{\boldsymbol{\eta}}) + \frac{\lambda}{2}\tilde{\boldsymbol{\eta}}^T\boldsymbol{\Sigma}^*\tilde{\boldsymbol{\eta}},$$

where $\boldsymbol{\Sigma}^*$ is the block diagonal matrix of $\boldsymbol{\Sigma}$ which satisfies $\|\boldsymbol{\omega}_k\|^2 = \boldsymbol{\eta}'_{\cdot k}\boldsymbol{\Sigma}\boldsymbol{\eta}_{\cdot k}$. We also have $L(\tilde{\boldsymbol{y}}^{(-i)}, \tilde{\boldsymbol{\eta}})$ replacing $\tilde{\boldsymbol{y}}$ by $\tilde{\boldsymbol{y}}^{(-i)}$.

By leave-one-out lemma of Craven and Wahba(1979), the minimizer of $L(\tilde{\boldsymbol{y}}^{(-i)}, \tilde{\boldsymbol{\eta}})$ with respect to $\tilde{\boldsymbol{\eta}}$ will be $\tilde{\boldsymbol{\eta}}^{(-i)}$. Thus, from the first order Taylor expansion, we obtain

$$\begin{aligned}
\mathbf{0} &= \frac{\partial L(\tilde{\boldsymbol{y}}^{(-i)}, \tilde{\boldsymbol{\eta}}^{(-i)})}{\partial \tilde{\boldsymbol{\eta}}^{(-i)}} \\
&= \frac{\partial L(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\eta}})}{\partial \tilde{\boldsymbol{\eta}}} + \frac{\partial^2 L(\tilde{\boldsymbol{y}}^0, \tilde{\boldsymbol{\eta}}^0)}{\partial \tilde{\boldsymbol{\eta}} \partial \tilde{\boldsymbol{\eta}}^T}(\tilde{\boldsymbol{\eta}}^{(-i)} - \tilde{\boldsymbol{\eta}}) + \frac{\partial^2 L(\tilde{\boldsymbol{y}}^0, \tilde{\boldsymbol{\eta}}^0)}{\partial \tilde{\boldsymbol{y}} \partial \tilde{\boldsymbol{\eta}}^T}(\tilde{\boldsymbol{y}}^{(-i)} - \tilde{\boldsymbol{y}}) \\
&= \mathbf{0} + \frac{\partial^2 L(\tilde{\boldsymbol{y}}^0, \tilde{\boldsymbol{\eta}}^0)}{\partial \tilde{\boldsymbol{\eta}} \tilde{\boldsymbol{\eta}}^T}(\tilde{\boldsymbol{\eta}}^{(-i)} - \tilde{\boldsymbol{\eta}}) - \boldsymbol{I}(\tilde{\boldsymbol{y}}^{(-i)} - \tilde{\boldsymbol{y}}) \\
&= (\boldsymbol{W}(\tilde{\boldsymbol{\eta}}^0) + n\lambda\boldsymbol{\Sigma}^*)(\tilde{\boldsymbol{\eta}}^{(-i)} - \tilde{\boldsymbol{\eta}}) - \boldsymbol{I}(\tilde{\boldsymbol{y}}^{(-i)} - \tilde{\boldsymbol{y}}),
\end{aligned}$$

where $(\tilde{\boldsymbol{y}}^0, \tilde{\boldsymbol{\eta}}^0)$ is a point between $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\eta}})$ and $(\tilde{\boldsymbol{y}}^{(-i)}, \tilde{\boldsymbol{\eta}}^{(-i)})$. Approximate it by $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\eta}})$, then we have

$$\tilde{\boldsymbol{\eta}} - \tilde{\boldsymbol{\eta}}^{(-i)} \simeq (\boldsymbol{W}(\tilde{\boldsymbol{\eta}}) + n\lambda\boldsymbol{\Sigma}^*)^{-1}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{y}}^{(-i)}) = (\boldsymbol{W} + n\lambda\boldsymbol{\Sigma}^*)^{-1}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{y}}^{(-i)}),$$

where

$$\begin{aligned}
\boldsymbol{W} &= \operatorname{diag}\begin{pmatrix} \boldsymbol{p}_{\cdot 1} \\ \vdots \\ \boldsymbol{p}_{\cdot m-1} \end{pmatrix} - \begin{bmatrix} \operatorname{diag}(\boldsymbol{p}_{\cdot 1}) \\ \vdots \\ \operatorname{diag}(\boldsymbol{p}_{\cdot m-1}) \end{bmatrix} \times [\operatorname{diag}(\boldsymbol{p}_{\cdot 1}), \ldots, \operatorname{diag}(\boldsymbol{p}_{\cdot m-1})] \\
&= \{\boldsymbol{W}^{kl}\}_{k,l=1}^{m-1} \text{ with } n \times n \text{ matrices } \boldsymbol{W}^{kl} \text{ as elements.}
\end{aligned}$$

Since $\boldsymbol{H} = (\boldsymbol{W} + n\lambda\Sigma^*)^{-1} = \{\boldsymbol{H}^{kl}\}_{k,l=1}^{m-1}$ with $n \times n$ matrices $\boldsymbol{H}^{kl}$ as elements, we have, for $i = 1, 2, \ldots, n$,

$$\boldsymbol{\eta}_{i\cdot} - \boldsymbol{\eta}_{i\cdot}^{(-i)} \simeq \boldsymbol{H}^{(i)}(\boldsymbol{y}_{i\cdot} - \boldsymbol{p}_{i\cdot}^{(-i)}), \tag{4.1}$$

where $\boldsymbol{H}^{(i)} = \{\boldsymbol{H}_{ii}^{kl}\}_{k,l=1}^{m-1}$ is a $(m-1) \times (m-1)$ matrix. By the first order Taylor expansion, we have

$$\begin{aligned} \boldsymbol{y}_{i\cdot} - \boldsymbol{p}_{i\cdot}^{(-i)} &= (\boldsymbol{y}_{i\cdot} - \boldsymbol{p}_{i\cdot}) + (\boldsymbol{p}_{i\cdot} - \boldsymbol{p}_{i\cdot}^{(-1)}) \\ &= (\boldsymbol{y}_{i\cdot} - \boldsymbol{p}_{i\cdot}) + \left(\frac{\partial b(\boldsymbol{\eta}_{i\cdot})}{\partial \boldsymbol{\eta}_{i\cdot}} - \frac{\partial b(\boldsymbol{\eta}_{i\cdot}^{(-i)})}{\partial \boldsymbol{\eta}_{i\cdot}} \right) \\ &\simeq (\boldsymbol{y}_{i\cdot} - \boldsymbol{p}_{i\cdot}) + \boldsymbol{W}^{(i)}(\boldsymbol{\eta}_{i\cdot} - \boldsymbol{\eta}_{i\cdot}^{(-i)}), \end{aligned} \tag{4.2}$$

where $\boldsymbol{W}^{(i)} = \{\boldsymbol{W}_{ii}^{kl}\}_{k,l=1}^{m-1}$ is a $(m-1) \times (m-1)$ matrix. From (4.1) and (4.2), we have $\boldsymbol{\eta}_{i\cdot} - \boldsymbol{\eta}_{i\cdot}^{(-i)} \simeq (\boldsymbol{I} - \boldsymbol{H}^{(i)}\boldsymbol{W}^{(i)})^{-1}\boldsymbol{H}^{(i)}(\boldsymbol{y}_{i\cdot} - \boldsymbol{p}_{i\cdot})$.

Hence, the ACV is given as follows:

$$\begin{aligned} \mathrm{ACV}(\boldsymbol{\theta}) &= \sum_{i=1}^{n} (-\boldsymbol{y}_{i\cdot}^T \boldsymbol{\eta}_{i\cdot} + b(\boldsymbol{\eta}_{i\cdot})) \\ &\quad + \sum_{i=1}^{n} \boldsymbol{y}_{i\cdot}^T (\boldsymbol{I} - \boldsymbol{H}^{(i)}\boldsymbol{W}^{(i)})^{-1}\boldsymbol{H}^{(i)}(\boldsymbol{y}_{i\cdot} - \boldsymbol{p}_{i\cdot}). \end{aligned} \tag{4.3}$$

5. Numerical Studies

We illustrate the performance of the proposed MKLR through three real data sets (New Thyroid, Wine and Glass) available from UCI Machine Learning Repository (Blake and Merz, 1998). These data sets are briefly described in Table 5.1. We repeat the procedure 50 times and compare the misclassification error rates for the MKLR, pairwise and one-vs-all nonlinear SVCs. The experiments are conducted in MATLAB environment over Pentium IV at 2.0GHz. The RBF kernel is used for these data sets. For each data set the optimal values of the kernel parameter σ^2 and the regularization parameter λ are obtained by ACV function for the MKLR and by GACV function (Wahba *et al.*, 1999) for pairwise and one-vs-all nonlinear SVCs.

The averages and boxplots of 50 misclassification error rates by three methods are shown in Table 5.2 and Figure 5.1, respectively. We can see that the MKLR provides better classification performance than pairwise and one-vs-all nonlinear SVCs for three data sets.

Table 5.1: Sizes of training and test data sets

	No. of classes	*No. of input variables*	*Training data*	*Test data*
New Thyroid	3	5	143	72
Wine	3	12	119	59
Glass	6	9	143	71

Table 5.2: Misclassification error rates

	New Thyroid	*Wine*	*Glass*
MKLR	0.0764	0.0267	0.3479
Pairwise SVC	0.3189	0.0354	0.6732
One-vs-all SVC	0.3389	0.0369	0.3893

Table 5.3: Average CPU times for training

	New Thyroid	*Wine*	*Glass*
MKLR	0.1716	0.0759	0.6675
Pairwise SVC	0.3000	0.3772	1.4734
One-vs-all SVC	0.8144	0.0882	0.3653

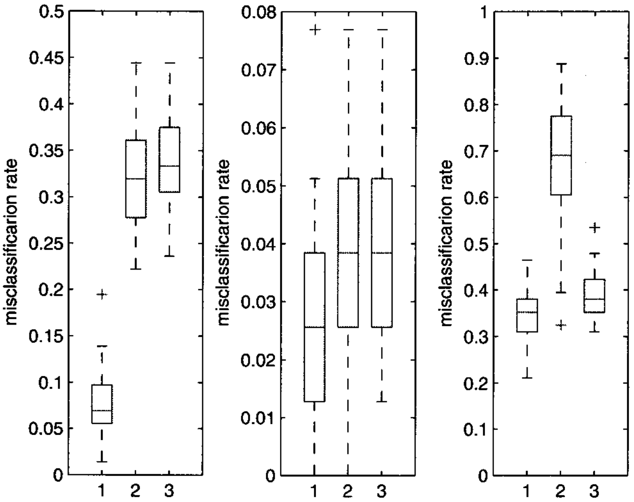

Figure 5.1: Boxplots of 50 misclassification error rates (1: MKLR, 2: Pairwise SVC, 3: One-vs-all SVC)

We also compare CPU times of the MKLR, pairwise and one-vs-all nonlinear SVCs computed by the built-in function of MATLAB. The averages of 50 CPU times in seconds for multiclassification by three methods are shown in Table 5.3, which indicate that the MKLR tends to be faster than pairwise nonlinear SVC.

6. Concluding Remarks

In this paper, we proposed the MKLR for multiclassification and obtained ACV function for the model selection. By using ACV function the model selection becomes easier and faster than that by a leave-one-out cross validation. Through three examples we showed that the MKLR provides the satisfying results and is attractive approach for multiclassification problem. We found that the MKLR takes less computing time than pairwise nonlinear SVC when being trained with fixed parameter values and then computing misclassification error rates for test data.

References

Blake, C. L. and Merz, C. J. (1998). UCI Repository of machine learning databases. University of California, Department of Information and Computer Science. Available from: http://www.ics.uci.edu/ mlearn/MLRepository.html.

Böhning, D. (1992). Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, **44**, 197–200.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematic*, **31**, 317–403.

Kimeldorf, G. S. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and its Applications*, **33**, 82–95.

Krishnapuram, B., Carin, L., Figueiredo, M. A. T. and Hartemink, A. J. (2005). Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE Ttransaction on Pattern Analysis and Machine Intelligence*, **27**, 957–968.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London*, **209**, 415–446.

Minka, T. (2003). A comparison of numerical optimizers for logistic regression. *Technical Report*, Department of Statistics, Carnegie Mellon University.

Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research*, **5**, 101–141.

Suykens, J. A. K. and Vandewalle, J. (1999). Multiclass least squares support vector machines, *Proceeding of the International Joint Conference on Neural Networks*, 900–903.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag, New York.

Vapnik, V. N. (1998). *Statistical Learning Theory.* Springer-Verlag, New York.

Wahba, G., Lin, Y., and Zhang, H. (1999). Generalized approximate cross validation for support vector machine, or, another way to look at margin-Like quantities. *Technical Report No. 1006*, University of Wisconsin.

Weston, J. and Watkins, C. (1998). Multi-class SVM. *Technical Report 98-04*, Royal Holloway University of London.

[Received July 2007, Accepted August 2007]