The Korean Communications in Statistics Vol. 15, No. 1, 2008, pp. 43–50

Sparse Multinomial Kernel Logistic Regression†

Jooyong Shim[1], Jongsig Bae[2], Changha Hwang[3]

Abstract

Multinomial logistic regression is a well known multiclass classification method in the field of statistical learning. More recently, the development of sparse multinomial logistic regression model has found application in microarray classification, where explicit identification of the most informative observations is of value. In this paper, we propose a sparse multinomial kernel logistic regression model, in which the sparsity arises from the use of a Laplacian prior and a fast exact algorithm is derived by employing a bound optimization approach. Experimental results are then presented to indicate the performance of the proposed procedure.

Keywords: Bound optimization; Laplacian regularization; multinomial logistic regression; sparsity; support vector machine.

1. Introduction

Multinomial logistic regression (MLR) is a popular method for multiclass classification problems. The output of a MLR model can be interpreted as a posterior estimate of the probability that an observation belongs to each of m disjoint classes. The probabilistic nature of the MLR model affords many practical advantages, such as the ability to accommodate unequal relative class frequencies in the training set or to apply an appropriate loss matrix in making predictions that minimize the expected risk. As a result, this model has been adopted in a diverse range of applications, including cancer classification and analysis of DNA binding sites.

† This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-311-C00246).

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

2) Professor, Department of Mathematics, Sungkyunkwan University, Suwon 440-746, Korea

3) Professor, Division of Information and Computer Science, Dankook University, Gyeonggido 448-160, Korea. Correspondence: chwang@dankook.ac.kr

Recently, a variety of methods have been explored that aim to introduce sparsity in supervised learning. The sparsity is desirable for the purpose of computational convenience, rather than as an aid to understanding the data. The sparsity arises from the use of a Laplacian prior. This family of algorithms includes the relevance vector machine (RVM) (Tipping, 2001), sparse online Gaussian processes (Csato and Opper, 2002) and the informative vector machine (IVM) (Lawrence *et al.,* 2003). The sparse MLR (SMLR) algorithm is one of such sparse methods that applies to multiclass and adopts a Laplacian prior to enforce sparseness (Krishnapuram *et al.,* 2005; Cawley *et al.,* 2006). These algorithms learn classifiers that are constructed as weighted linear combinations of basis functions, where the weights are estimated in the presence of training data. In many of these algorithms, the set of permitted basis functions may be the original input vectors themselves, some nonlinear transformation of those vectors or even kernels centered on the training samples. In the latter case, the learned classifier will be similar in flavor to a support vector machine (SVM) by Vapnik (1995).

In this paper we propose a kernel variant of SMLR of Krishnapuram *et al.* (2005), which is called SMKLR. The SMKLR is derived by employing a bound optimization approach based on Laplacian prior as in Krishnapuram *et al.* (2005). Accordingly, we derive a fast exact algorithm for learning SMKLR that scale favorably in the number of training samples, making it applicable to large data sets. We compare SMKLR with RVM and SVM over a range of benchmark data sets in terms of misclassification rates and numbers of basis kernels retained.

2. Multinomial Kernel Logistic Regression

In this section, we provide a brief description of the MKLR model. 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 $m \times 1$ 0/1 valued vector $\boldsymbol{y} = (y_1, \ldots, y_m)^T$, where $y_k = 1$ and all other coordinates are 0. MLR 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 $dm \times 1$ vector $\boldsymbol{\omega} = (\boldsymbol{\omega}_1^T, \ldots, \boldsymbol{\omega}_m^T)^T$, where $\boldsymbol{\omega}_k$ is the $d \times 1$ 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 $d(m-1) \times 1$ 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:

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

A nonlinear form of MLR, known as MKLR, can be obtained via the so-called "kernel trick", whereby a conventional MLR model is constructed in a high dimensional feature space induced by a Mercer (1909)'s kernel. More formally, given training data, $\mathcal{D}$, 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 MLR model constructed in the feature space is given as follows:

$$\ell(\boldsymbol{\eta}) = -\sum_{i=1}^{n}\sum_{k=1}^{m-1} y_{ik}\eta_{ik} + \sum_{i=1}^{n}\log\left(1 + \sum_{k=1}^{m-1}\exp(\eta_{ik})\right), \tag{2.4}$$

where $\eta_{ik} = \boldsymbol{\omega}_k^T\boldsymbol{\phi}(\boldsymbol{x}_i)$. The representation theorem (Kimeldorf and Wahba, 1971) guarantees that the minimizer of the negative log-likelihood (2.4) 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 $n(m-1) \times 1$ vector $\boldsymbol{\alpha}$ to minimize

$$\ell(\boldsymbol{\alpha}) = -\sum_{i=1}^{n}\sum_{k=1}^{m-1} y_{ik}\boldsymbol{K}_i^T\boldsymbol{\alpha}_k + \sum_{i=1}^{n}\log\left(1+\sum_{k=1}^{m-1}\exp\left(\boldsymbol{K}_i^T\boldsymbol{\alpha}_k\right)\right), \tag{2.5}$$

where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_{m-1}^T)^T$. The minimization 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. See for details Minka (2003).

3. Sparse Multinomial Kernel Logistic Regression

The main drawback of the MKLR is that all training vectors are involved in the final solution which is not acceptable for large data sets like gene selection tasks. The SMKLR could be achieved here if we utilize a bound optimization approach based on Laplacian prior.

The maximum likelihood estimate (MLE) $\boldsymbol{\alpha}$ minimizing (2.5) generally leads severe overfitting and so we are motivated to adopt a maximum a posteriori (MAP) estimate or penalized MLE,

$$\boldsymbol{\alpha}_{MAP} = \arg\min_{\boldsymbol{\alpha}}\left[\ell(\boldsymbol{\alpha}) - \sum_{k=1}^{m-1}\log p(\boldsymbol{\alpha}_k)\right], \tag{3.1}$$

where $p(\boldsymbol{\alpha}_k)$ is the Laplacian prior on $\boldsymbol{\alpha}_k$, which means that

$$p(\boldsymbol{\alpha}_k) \propto \exp(-\lambda\|\boldsymbol{\alpha}_k\|_1), \tag{3.2}$$

where $\|\boldsymbol{a}_k\|_1 = \sum_{i=1}^{n}|\alpha_{ki}|$ denotes L_1 norm and λ acts as a tunable penalty parameter. Then the penalized negative log-likelihood function of the MKLR model can be rewritten as follows:

$$L(\boldsymbol{\alpha}) = \ell(\boldsymbol{\alpha}) + \lambda\sum_{k=1}^{m-1}\|\boldsymbol{\alpha}_k\|_1. \tag{3.3}$$

The inclusion of a Laplacian prior does not allow the use of the classical IRWLS method. The bound optimization approach provides us with a tool to attack this optimization problem. The key concept in bound optimization is that $L(\boldsymbol{\alpha})$ is optimized by iteratively maximizing a surrogate function Q as follows:

$$\boldsymbol{\alpha}^{(t+1)} = \arg\min_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}) \tag{3.4}$$

A valid surrogate function can be

$$Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)}) = \ell(\boldsymbol{\alpha}) + \frac{\lambda}{2}\sum_{k=1}^{m-1}\sum_{i=1}^{n}\left(\frac{\alpha_{ki}^2}{|\alpha_{ki}^{(t)}|} + |\alpha_{ki}^{(t)}|\right). \tag{3.5}$$

Now, $Q(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(t)})$ is differentiable with respect to $\boldsymbol{\alpha}$. Thus, we can use IRWLS procedure to update $\boldsymbol{\alpha}$. The details are illustrated in Krishnapuram *et al.* (2005).

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, the minimization of this surrogate function leads to the update equation

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - (\boldsymbol{H}^{(t)} + \lambda\boldsymbol{L}^{(t)})^{-1}\boldsymbol{G}^{(t)}, \tag{3.6}$$

where $\boldsymbol{K}^* = \mathrm{diag}(\boldsymbol{K},\ldots,\boldsymbol{K})$, $\boldsymbol{L}^{(t)}$ is the $(m-1)n \times (m-1)n$ diagonal matrix consisting of $|\alpha_{ki}^{(t)}|^{-1}$, and $\boldsymbol{H}^{(t)}$ and $\boldsymbol{G}^{(t)}$ are defined as

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

$$\boldsymbol{G}^{(t)} = \boldsymbol{K}^*\left[-\begin{bmatrix}\boldsymbol{y}_{\cdot 1}\\ \vdots \\ \boldsymbol{y}_{\cdot m-1}\end{bmatrix} + \begin{bmatrix}\boldsymbol{p}_{\cdot 1}^{(t)}\\ \vdots \\ \boldsymbol{p}_{\cdot m-1}^{(t)}\end{bmatrix}\right] + \lambda\boldsymbol{L}^{(t)}\begin{bmatrix}\boldsymbol{\alpha}_1^{(t)}\\ \vdots \\ \boldsymbol{\alpha}_{m-1}^{(t)}\end{bmatrix}.$$

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

$$\boldsymbol{H}^{(t)} = \sum_{i=1}^{n}(\mathrm{diag}(\boldsymbol{p}_{i\cdot}^{(t)}) - \boldsymbol{p}_{i\cdot}^{(t)}\boldsymbol{p}_{i\cdot}^{(t)T}) \otimes \boldsymbol{K}_i\boldsymbol{K}_i^T, \tag{3.7}$$

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 $\boldsymbol{\alpha}$,

$$\boldsymbol{H}^{(t)} \le \frac{1}{2}(\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T/m) \otimes \sum_{i=1}^{n}\boldsymbol{K}_i\boldsymbol{K}_i^T \equiv \boldsymbol{B},$$

where $\boldsymbol{I}$ is an identity matrix and $\boldsymbol{1} = (1,\ldots,1)^T$. Note $\boldsymbol{H}^{(t)} \le \boldsymbol{B}$ means $\boldsymbol{H}^{(t)} - \boldsymbol{B}$ is negative semidefinite. Thus, using the bound optimization technique, we have a simple IRWLS procedure for updating $\boldsymbol{\alpha}$,

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - (\boldsymbol{B} + \lambda\boldsymbol{L}^{(t)})^{-1}\boldsymbol{G}^{(t)}. \tag{3.8}$$

Table 4.1: Characteristics of six real data sets

	Iris	*New Thyroid*	*Wine*	*Leukemia*	*Lymphoma*	*Brain*
No. of classes	3	3	3	3	3	5
No. of input variables	4	5	13	3572(20)	4027(20)	5598(20)
Size of training sample	100	143	119	48	41	28
Size of test sample	50	72	59	24	21	14
Size of total sample	150	215	178	72	62	42

Using upper bound matrix $\boldsymbol{B}$, we do not need to compute Hessian at each iteration, which yields a fast algorithm for the SMKLR. It is now possible to perform exact MAP MKLR under a Laplacian prior for the same cost as the original IRWLS algorithm for ML estimation.

4. Numerical Studies

This section illustrates how well SMKLR works for the sparseness and the classification using the six publicly available real data sets. The characteristics of these data sets are briefly described in Table 4.1.

Each data set is randomly divided into a training sample (67%) and a test sample (33%). The number in parenthesis represents the number of input variables actually used in the study. It is noted that RVM does not work properly for the case where input variables are more than observations. It is determined by the recursive feature addition (RFA) with ranking criteria relevant to 5-fold cross validation. We compare three sparse algorithms, *i.e.,* SMKLR, one-vs-all SVM and one-vs-all RVM in terms of the misclassification rates and the numbers of basis kernels retained. Both SVM and RVM are typical sparse algorithms. We use here one-vs-all version of SVM and RVM since the most widely used implementation is the one-vs-all method (Rifkin and Klautau, 2004). The Gaussian kernel is used for these data sets. The procedure was repeated 50 times. For each data set the optimal values of the kernel parameter σ^2 and the penalty parameter λ are determined by 5-fold cross validation. The experiments are conducted in MATLAB environment over Pentium IV at 2.0GHz.

The averages of 50 misclassification rates by three methods are shown in Table 4.2 together with the average numbers of basis kernels retained. The average numbers are given in parentheses. As seen from Table 4.2, the SMKLR provides overall better classification performance than one-vs-all SVM and RVM for six data sets. The SMKLR retains less basis kernels than one-vs-all SVM and RVM

Table 4.2: Average misclassification rates and average numbers of basis kernels retained

	Iris	*New Thyroid*	*Wine*	*Leukemia*	*Lymphoma*	*Brain*
SMKLR	0.0492	0.0947	0.0285	0.2392	0.0286	0.3543
	(31.88)	(80.71)	(16.60)	(29.72)	(24.29)	(9.80)
one-vs-all SVM	0.0568	0.3439	0.0380	0.3317	0.0829	0.4229
	(15.24)	(143.0)	(31.43)	(36.59)	(32.77)	(15.03)
one-vs-all RVM	0.0556	0.0811	0.0502	0.3725	0.0286	0.4729
	(6.13)	(103.77)	(3.69)	(47.39)	(41.0)	(15.79)

for the rest four data sets except Iris and Wine data sets. We realize that the SMKLR provides the satisfying results regarding the classification accuracy and the sparseness.

5. Conclusions

In this paper, we have proposed the SMKLR for learning sparse multiclassification. In fact, sparseness is essential to achieve good generalization capabilities of MKLR and can be enforced by using heavy tailed priors on the weights of the linear combination of kernel functions. The SMKLR adopts a Laplacian prior to enforce sparseness. Our numerical studies with six data sets demonstrate that the SMKLR provides the satisfying results regarding the classification accuracy and the sparseness, and thus is attractive approach for multiclassification problems. We also have found that the SMKLR takes less computing time than one-vs-all RVM when being trained with fixed parameter values. Its applicability to large data sets is still a delicate task from the computational point of view.

References

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

Cawley, G. C., Talbot, N. L. C. and Girolami, M. (2006). Sparse multinomial logistic regression via Bayesian L1 regularisation. *Advances in Neural Information Processing Systems,* **18**, 609–616.

Csato, L. and Opper, M. (2002). Sparse online Gaussian processes. *Neural Computation,* **14**, 641–668.

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.

Lawrence, N. D., Seeger, M. and Herbrich, R. (2003). Fast sparse Gaussian process methods: the informative vector machine. *Advances in Neural Information Processing Systems,* **15**, 609–616.

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.

Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research,* **1**, 211–244.

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

[Received July 2007, Accepted October 2007]