Journal of the Korean Data & Information Science Society
2010, **21**(3), 561–567

한국데이터정보과학회지

Fixed size LS-SVM for multiclassification problems of large data sets [†]

Hyung Tae Hwang[1]

[1]Department of Statistics, Dankook University

Received 12 March 2010, revised 23 April 2010, accepted 30 April 2010

요 약

Multiclassification is typically performed using voting scheme methods based on combining a set of binary classifications. In this paper we use multiclassification method with a hat matrix of least squares support vector machine (LS-SVM), which can be regarded as the revised one-against-all method. To tackle multiclass problems for large data, we use the Nyström approximation and the quadratic Renyi entropy with estimation in the primal space such as used in fixed size LS-SVM. For the selection of hyperparameters, generalized cross validation techniques are employed. Experimental results are then presented to indicate the performance of the proposed procedure.

Keywords: Fixed size least squares support vector machine, generalized cross validation, multiclass, Nyström approximation, quadratic Renyi entropy.

1. Introduction

Many real applications consist of multiclass problems. Support vector machine (SVM) was originally designed by Vapnik (1995) for binary classification. SVM is gaining popularity due to many attractive features and promising empirical performance. Extending it to multiclass problems is an ongoing research issue. There are commonly two types of multiclass extensions for SVM. One is the composition type methods built on a series of binary classification methods such as the one-against-one, one-against-all and error correcting output codes (Allwein *et al.*, 2000; Dietterich and Bakiri, 1995), and the other is the single machine type methods, which attempt to construct a multiclass classifier by solving a single optimization problem (Vapnik, 1998; Weston and Watkins, 1998; Lee *et al.*, 2001). There is no substantial agreement on which method is the best one for the multiclass problem (Rifkin and Klautau, 2004).

Despite of many successful application of SVM in classification and regression problem, training an SVM requires to solve a quadratic program (QP) problem. The QP is to optimize a quadratic function over a polyhedron, defined by linear equations and/or inequalities, which is time memory expensive. Suykens and Vanderwalle (1995) proposed LS-SVM for binary classification. Its solution is given by a linear equation system instead of a QP

[†] The present research was conducted by the research fund of Dankook University in 2009.
[1] Department of Statistics, Dankook University, Gyeonggido 448-701, Korea.
Email: hthwang@dankook.ac.kr

problem. LS-SVM keeps explicit primal-dual formulations which has lots of advantages. Suykens and Vanderwalle (1999) proposed an extension of LS-SVM to the multiclass case. In stead of QP, some other techniques such as iterative reweighted least squares technique and Newton-Raphson method have been used in kernel machines. See Hwang (2007, 2008), Shim and Seok (2008) and Shim *et al.* (2009) for details.

Espinoza *et al.* (2005) proposed the fixed size LS-SVM for large scale regression problems by using the sparse approximation of nonlinear feature mapping function induced by kernel function, whose computation is based on the Nyström approximation (Williams and Seeger, 2001) and the quadratic Renyi entropy (Girolami, 2003).

In this paper we propose an LS-SVM solving multiclass problems of large data sets with fixed size LS-SVM regression. This method implements one-against-all scheme which is as accurate as any other approach, which can be considered as the extension of the multiclass LS-SVM (Shim *et al.*, 2008) to the large data problem. We also derive the generalized cross validation (GCV) function to select the hyperparameters which affect the performance of the proposed multiclass LS-SVM method. The rest of paper is organized as follows. In Section 2 we briefly illustrate LS-SVM regression and its relationship to LS-SVM classification. In Section 3 we describe LS-SVM regression for multiclassification. In Section 4 we propose multiclass LS-SVM for large data and GCV function for model selection. In Section 5 we perform the numerical studies with real data sets. In Section 6 we give the conclusions.

2. LS-SVM

LS-SVMs have been successfully applied to static problems like classification and function estimation. LS-SVMs have been extended to recurrent models and used in optimal control problems. See for further details Suykens and Vanderwalle (1995, 1999) and Suykens (2001). In this section we review some basic idea of LS-SVM regression. We also illustrate LS-SVM classification is actually equivalent to LS-SVM regression in binary classification case.

The LS-SVM model for regression estimation has the following representation in feature space

$$y(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{\Phi}(\boldsymbol{x}) + b, \tag{2.1}$$

where $\boldsymbol{x} \in R^d, y \in R$, and $\boldsymbol{w} \in R^{d_f}$ is a weight vector corresponding to $\boldsymbol{\Phi}(\boldsymbol{x})$. The use of the nonlinear mapping $\boldsymbol{\Phi}(\cdot)$ is similar to the classifier case.

Given a training data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ with each input $\boldsymbol{x}_i \in R^d$ and corresponding output $y_i \in R$, we consider the following optimization problem in primal weight space:

$$L(\boldsymbol{w}, b, \boldsymbol{e}) = \frac{1}{2}\boldsymbol{w}^t\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{n} e_i^2 \tag{2.2}$$

subject to equality constraints

$$y_i = \boldsymbol{w}^t \boldsymbol{\Phi}(\boldsymbol{x}_i) + b + e_i, \ i = 1, \cdots, n. \tag{2.3}$$

The cost function with squared error and regularization corresponds to a form of ridge regression. To find minimizers of the objective function, we can construct the Lagrangian

function as follows:

$$L(\boldsymbol{w}, b, \boldsymbol{e}; \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^t\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{n} e_i^2 - \sum_{i=1}^{n}\alpha_i(\boldsymbol{w}^t\boldsymbol{\Phi}(\boldsymbol{x}_i) + b + e_i - y_i), \tag{2.4}$$

where α_i's are the Lagrange multipliers. Then, the conditions for optimality are given by

$$\begin{aligned}
&\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{0} \rightarrow \boldsymbol{w} = \sum_{i=1}^{n}\alpha_i\boldsymbol{\Phi}(\boldsymbol{x}_i)\\
&\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{n}\alpha_i = 0\\
&\frac{\partial L}{\partial e_i} = 0 \rightarrow e_i = \frac{1}{\gamma}\alpha_i, \quad i = 1, \cdots, n\\
&\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow y_i - b - \boldsymbol{w}^t\boldsymbol{\Phi}(\boldsymbol{x}_i) - e_i = 0, \quad i = 1, \cdots, n
\end{aligned} \tag{2.5}$$

After eliminating e_i and $\boldsymbol{w}$, we could have the solution by the following linear equations

$$\begin{bmatrix} \boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}_n & \boldsymbol{1}_n \\ \boldsymbol{1}_n^t & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}, \tag{2.6}$$

where $\boldsymbol{1}_n$ is the $n \times 1$ vector of ones and $\boldsymbol{K}$ is the $n \times n$ kernel matrix with elements $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\Phi}(\boldsymbol{x}_i)^t\boldsymbol{\Phi}(\boldsymbol{x}_j)$, $i, j = 1, \cdots, n$.

Solving the linear equation (2.6), the optimal bias b and Lagrange multipliers α_i's are obtained, and then the optimal regression function for a test data point $\boldsymbol{x}_t^*$ is obtained as

$$\hat{y}(\boldsymbol{x}_t^*) = \sum_{i=1}^{n} K(\boldsymbol{x}_t^*, \boldsymbol{x}_i)\alpha_i + b. \tag{2.7}$$

Note that it can be easily shown that Lagrange multipliers of LS-SVM for binary classification are identical to Lagrange multipliers of LS-SVM for regression obtained from equation (2.6), when class labels are -1 and 1. That is, if $\boldsymbol{y}$ consists of class labels -1 and 1, $\hat{\boldsymbol{y}}$ obtained by LS-SVMs for regression and classification are identical. Thus, for the binary classification, each observation of the test data can be classified into either class according to the sign of $\hat{y}(\boldsymbol{x}_t^*)$ in equation (2.7) for $t = 1, \cdots, n_t$. See for details Shim *et al.* (2008). We use LS-SVM for regression, instead of LS-SVM for classification, to approximate the cross validation function of multiclass LS-SVM.

3. Multiclass LS-SVM

3.1. One-against-all multiclass LS-SVM

In this section we give simple overview on multiclassification by LS-SVM using one-against-all method (Shim *et al.*, 2008) which uses the fact that LS-SVM classification is equivalent to LS-SVM regression for binary classification case.

Let the training data set be denoted by $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ with each input vector $\boldsymbol{x}_i \in R^d$ and the class label $y_i \in \{1, 2, \cdots, m\}$, where m is the number of classes. One-against-all multiclass

LS-SVM regression constructs m binary LS-SVM regressor, each of which separates one class from all the rest. The jth LS-SVM regressor is trained with all the training examples of the jth class with positive labels, and all the others with negative labels. Thus, for one-against-all multiclass LS-SVM regression, we transform $\boldsymbol{y}$ into $n \times m$ matrix $\boldsymbol{Y}$ which consists of -1 and 1 such that $Y_{ij} = 1$ and $Y_{ik} = -1$ for $j \neq k$ implies ith example belongs to the jth class. We have m LS-SVM regressors for binary classification with $\{\boldsymbol{x}_i, Y_{ij}\}_{i=1}^n$ for $j = 1, \cdots, m$.

From the linear equation system

$$\begin{bmatrix} \boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}_n & \boldsymbol{1}_n \\ \boldsymbol{1}_n^t & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}^j \\ b^j \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{\cdot j} \\ 0 \end{bmatrix}, \tag{3.1}$$

the optimal bias b^j and Lagrange multipliers α_i^j's are obtained. Here $\boldsymbol{Y}_{\cdot j}$ is the jth column of $\boldsymbol{Y}$.

For a test data point $\boldsymbol{x}_t^*$, we have

$$\hat{Y}_{tj}(\boldsymbol{x}_t^*) = \sum_{i=1}^{n} K(\boldsymbol{x}_t^*, \boldsymbol{x}_i)\alpha_i^j + b^j, \quad \text{for} \quad t = 1, \cdots, n_t. \tag{3.2}$$

Thus, if $\hat{Y}_{tj}(\boldsymbol{x}_t^*) > 0$ and $\hat{Y}_{tk}(\boldsymbol{x}_t^*) < 0$ for $k \neq j$, then the test data point $\boldsymbol{x}_t^*$ is classified into the jth class for $t = 1, \cdots, n_t$.

3.2. Multiclass LS-SVM using hat matrix

In this section we use the hat matrix for the test data to avoid solving linear equations in equation (3.1). We denote the sets of $\boldsymbol{x}_i$'s and $\boldsymbol{x}_t^*$'s by $\mathcal{X}$ and $\mathcal{X}_t$, respectively. For convenience we will use some notations such as $\hat{\boldsymbol{Y}}_{\cdot j}(\mathcal{X}_t), \hat{\boldsymbol{Y}}(\mathcal{X}_t), \boldsymbol{S}(\mathcal{X}_t, \mathcal{X}), \hat{\boldsymbol{\Phi}}(\mathcal{X})$ to denote vectors or matrices constructed based on $\mathcal{X}$ and $\mathcal{X}_t$. For $\boldsymbol{K}_t = \{K_{tl}\}$ with $K_{tl} = K(\boldsymbol{x}_t^*, \boldsymbol{x}_l), t = 1, \cdots, n_t, l = 1, \cdots, n$, we can rewrite equation (3.2) as follows:

$$\begin{aligned} \hat{\boldsymbol{Y}}_{\cdot j}(\mathcal{X}_t) &= \boldsymbol{K}_t\boldsymbol{\alpha}^j + b^j\boldsymbol{1}_{n_t} = [\boldsymbol{K}_t, \boldsymbol{1}_{n_t}] \begin{bmatrix} \boldsymbol{\alpha}^j \\ b^j \end{bmatrix} = [\boldsymbol{K}_t, \boldsymbol{1}_{n_t}] \begin{bmatrix} \boldsymbol{S}_{11} & \boldsymbol{s}_{12} \\ \boldsymbol{s}_{21} & s_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{Y}_{\cdot j} \\ 0 \end{bmatrix} \\ &= (\boldsymbol{K}_t\boldsymbol{S}_{11} + \boldsymbol{1}_{n_t}\boldsymbol{s}_{21})\boldsymbol{Y}_{\cdot j} = \boldsymbol{S}(\mathcal{X}_t, \mathcal{X})\boldsymbol{Y}_{\cdot j}, \quad j = 1, \cdots, m, \end{aligned} \tag{3.3}$$

where $\boldsymbol{S}_{11}$ consists of first n rows and first n columns of inverse of the leftmost matrix in equation (3.1) and $\boldsymbol{s}_{21}$ consists of the last row and first n columns of inverse of the leftmost matrix in equation (3.1). Note that $\hat{\boldsymbol{Y}}_{\cdot j}(\mathcal{X}_t) = (\hat{Y}_{1j}(\boldsymbol{x}_1^*), \cdots, \hat{Y}_{n_t j}(\boldsymbol{x}_{n_t}^*))^t$.

Since $\boldsymbol{S}(\mathcal{X}_t, \mathcal{X})$ does not depend on $\boldsymbol{Y}_{\cdot j}$, we can write the predicted $\hat{\boldsymbol{Y}}(\mathcal{X}_t)$ as,

$$\hat{\boldsymbol{Y}}(\mathcal{X}_t) = \boldsymbol{S}(\mathcal{X}_t, \mathcal{X})\boldsymbol{Y}, \tag{3.4}$$

where $\hat{\boldsymbol{Y}}(\mathcal{X}_t) = (\hat{\boldsymbol{Y}}_{\cdot 1}(\mathcal{X}_t), \cdots, \hat{\boldsymbol{Y}}_{\cdot m}(\mathcal{X}_t))$ is an $n_t \times m$ matrix. Thus, we need not to solve m linear equations in equation (3.1) but once for given $\boldsymbol{S}(\mathcal{X}_t, \mathcal{X})$.

4. Multiclass LS-SVM for large data

4.1. Multiclassification by fixed size LS-SVM

For large data (large n), it is not possible to obtain the estimator of parameter vector from equation (3.1) since it is not computable to find the inverse of a $(n+1) \times (n+1)$ matrix.

We consider the extension of the multiclass LS-SVM to the large data problem. In fixed size LS-SVM (Espinoza *et al.*, 2005) the Nyström approximation and the quadratic Renyi entropy are used to obtain the approximate of the feature mapping function, $\hat{\boldsymbol{\Phi}}(\mathcal{X})$, which is a $n \times n_s$ matrix with $n_s(<< n)$ the specified number of support vectors,

$$\hat{\boldsymbol{\Phi}}(\mathcal{X}) \propto \boldsymbol{K}(\mathcal{X}, \mathcal{X}_s)\boldsymbol{E}\boldsymbol{D}^{-1/2}, \tag{4.1}$$

where $\mathcal{X}_s$ are the set of support vectors chosen by using the quadratic Renyi entropy, $\boldsymbol{E}$ is the matrix composed of eigenvectors of $\boldsymbol{K}(\mathcal{X}_s, \mathcal{X}_s)$ and $\boldsymbol{D}$ is the diagonal matrix of eigenvalues of $\boldsymbol{K}(\mathcal{X}_s, \mathcal{X}_s)$. From the approximate of the feature mapping function, $\hat{\boldsymbol{\Phi}}(\mathcal{X})$, computed based on the subsample, the model is estimated in the primal space.

In the multiclass LS-SVM for large data we have the ridge estimate of the jth column of $\boldsymbol{Y}$, $j = 1, \cdots, m$, from (2.3) as

$$\begin{aligned}\hat{\boldsymbol{Y}}_{\cdot j} &= (\hat{\boldsymbol{\Phi}}(\mathcal{X}), \mathbf{1}_n) \begin{bmatrix} \hat{\boldsymbol{\Phi}}(\mathcal{X})^t\hat{\boldsymbol{\Phi}}(\mathcal{X}) + \frac{1}{\gamma}\boldsymbol{I} & \hat{\boldsymbol{\Phi}}(\mathcal{X})^t\mathbf{1}_n \\ \mathbf{1}_n^t\hat{\boldsymbol{\Phi}}(\mathcal{X}) & n \end{bmatrix}^{-1} \begin{bmatrix} \hat{\boldsymbol{\Phi}}(\mathcal{X})^t \\ \mathbf{1}^t \end{bmatrix} \boldsymbol{Y}_{\cdot j} \\ &= \boldsymbol{S}(\mathcal{X}, \mathcal{X})\boldsymbol{Y}_{\cdot j}, \end{aligned} \tag{4.2}$$

where $\boldsymbol{Y}_{\cdot j}$ is the jth column of $\boldsymbol{Y}$. For the test data set $\mathcal{X}_t$,

$$\begin{aligned}\hat{\boldsymbol{Y}}_{\cdot j}(\mathcal{X}_t) &= (\hat{\boldsymbol{\Phi}}(\mathcal{X}_t), \mathbf{1}_{n_t}) \begin{bmatrix} \hat{\boldsymbol{\Phi}}(\mathcal{X})^t\hat{\boldsymbol{\Phi}}(\mathcal{X}) + \frac{1}{\gamma}\boldsymbol{I} & \hat{\boldsymbol{\Phi}}(\boldsymbol{x})^t\mathbf{1}_{n_t} \\ \mathbf{1}_{n_t}^t\hat{\boldsymbol{\Phi}}(\boldsymbol{x}) & n \end{bmatrix}^{-1} \begin{bmatrix} \hat{\boldsymbol{\Phi}}(\mathcal{X})^t \\ \mathbf{1}_{n_t}^t \end{bmatrix} \boldsymbol{Y}_{\cdot j} \\ &= \boldsymbol{S}(\mathcal{X}_t, \mathcal{X})\boldsymbol{Y}_{\cdot j}. \end{aligned} \tag{4.3}$$

4.2. Model selection

The functional structure of multiclass LS-SVM is characterized by hyperparameters, the regularization parameter and the kernel parameters. To select the parameters of multiclass LS-SVM, we define the cross validation (CV) function as follows:

$$CV(\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_{im_i} - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})\right)^2, \tag{4.4}$$

where $\boldsymbol{\lambda}$ is the set of hyperparameters and $\hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})$ is the predicted value of Y_{im_i} obtained from the data without ith observation. Here m_i is the column number of the ith row of $\boldsymbol{Y}$ such that $Y_{im_i} = 1$, which implies that the ith observation belongs to the m_ith class. The CV can be rewritten as

$$CV(\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})\right)^2. \tag{4.5}$$

Since for each candidate of hyperparameters, $\hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})$ for $i = 1, \cdots, n$ should be evaluated, selecting parameters using CV function is computationally formidable.

By leaving-out-one lemma (Kimeldorf and Wahba, 1971) and the first order Taylor expansion, we have

$$Y_{im_i} - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda}) \approx \frac{Y_{im_i} - \hat{Y}_{im_i}}{1 - \frac{\partial\hat{Y}_{im_i}}{\partial Y_{im_i}}} \quad \text{and} \quad \hat{Y}_{im_i} = \boldsymbol{S}_{i\cdot}\boldsymbol{Y}_{\cdot m_i}, \tag{4.6}$$

where $\boldsymbol{S}_{i\cdot}$ for $i = 1, \cdots, n$ is the ith row of the hat matrix $\boldsymbol{S}(\mathcal{X}, \mathcal{X})$ in equation (4.2). Then the ordinary cross validation (OCV) function can be obtained as

$$OCV(\boldsymbol{\lambda}) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1 - \hat{Y}_{im_i}(\boldsymbol{\lambda})}{1 - s_{ii}(\boldsymbol{\lambda})} \right)^2. \tag{4.7}$$

where s_{ii} for $i = 1, \cdots, n$, is the ith diagonal element of the hat matrix $\boldsymbol{S}(\mathcal{X}, \mathcal{X})$.

By replacing s_{ii}'s in equation (4.7) with their average $\mathrm{tr}(\boldsymbol{S}(\mathcal{X}, \mathcal{X}))/n$, the generalized cross validation (GCV) function can be then obtained as follows:

$$GCV(\boldsymbol{\lambda}) = \frac{n \sum_{i=1}^{n} \left(1 - \hat{Y}_{im_i}(\boldsymbol{\lambda})\right)^2}{\left(n - \mathrm{tr}(\boldsymbol{S}(\mathcal{X}, \mathcal{X}))\right)^2}. \tag{4.8}$$

5. Numerical studies

We illustrate the performance of the proposed procedure through the data sets available from UCI Machine Learning Depository (http://kdd.ics.uci.edu), which are the vowel-recognition data set extracted from the letter-recognition and the satellite data set. The Gaussian kernel is used in these examples, which is defined as

$$K(\boldsymbol{x}_k, \boldsymbol{x}_l) = e^{-\frac{\|\boldsymbol{x}_k - \boldsymbol{x}_l\|^2}{\sigma^2}}, \tag{5.1}$$

where σ^2 is the kernel parameter.

The vowel-recognition data set of 5 classes (5 vowels) has 16 variables and 3491 observations. We randomly divided it into training data set of size 2000 and test data set of size 1491. To illustrate the effect of prespecified number of support vectors, multiclass LS-SVM is tested for n_s = 20, 40, 60, 80 (1% $\sim$ 4% of training data). The optimal values of (γ, σ^2) are chosen from GCV function as (1,57), (1,42), (1,48) and (1,32), respectively.

The satellite data set of 6 classes has 16 variables, 4435 observations in the training data set and 2000 observations in the test data set. Multiclass LS-SVM is tested for n_s = 45, 90, 135, 180(1% $\sim$ 4% of training data). The optimal values of (γ, σ^2) are chosen from GCV function as (1,66), (1,38), (1,31) and (1,33), respectively. Table 5.1 shows the performance of the proposed method on the test data sets for various numbers of support vectors. From the results of two examples we can see that the larger numbers of support vectors provide the smaller misclassification error rates.

Table 5.1 Misclassification error rates of the test data sets for various numbers of support vectors.

Vowel data		Satellite data	
support vectors	error rate	support vectors	error rate
20	0.0885	45	0.1635
40	0.0570	90	0.1365
60	0.0376	135	0.1245
80	0.0335	180	0.1195

6. Conclusions

In this paper, we dealt with a multiclassification for a large scale data problem, in which we showed that it is possible to catch the structure of whole data with a small portion of data(support vectors). Through the examples we showed that the proposed method derives the satisfying results by using small number of support vectors and the accuracy increases with larger numbers of support vectors. We also found that the results are not much affected by the value of the regularization parameter but the kernel parameter.

References

Allwein, E. L., Schapire, R. E. and Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, **1**, 113-141.

Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, **2**, 263-286.

Espinoza, M., Suykens, J. A. K. and De Moor, B. (2005). Load forecasting using least squares support vector machines. *Lecture Notes in Computer Science*, **3512**, 1018-1026.

Girolami, M. (2003). Orthogonal series density estimation and kernel eigenvalue problem. *Neural Computation*, **14**, 669-688.

Hwang, C. (2007). Kernel machine for Poisson regression. *Journal of Korean Data & Information Science Society*, **18**, 767-772.

Hwang, C. (2008). Mixed effects kernel binomial regression. *Journal of Korean Data & Information Science Society*, **19**, 1327-1334.

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

Lee, Y. , Lin, Y. and Wahba, G. (2001). *Multicategory support vector machines*, Technical Report 1043, Department of Statistics, University of Wisconsin.

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

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

Shim, J., Bae, J. and Hwang, C. (2008). Multiclass classification via LS-SVR. *Communications of the Korean Statistical Society*, **15**, 441-450.

Shim, J., Park, H. and Hwang, C. (2009). A kernel machine for estimation of mean and volatility functions. *Journal of Korean Data & Information Science Society*, **20**, 905-912.

Shim, J. and Seok, K. H. (2008). Kernel poisson regression for longitudinal data. *Journal of Korean Data & Information Science Society*, **19**, 1353-1360.

Suykens, J. A. K. and Vanderwalle, J. (1995). Least square support vector machine classifier. *Neural Processing Letters*, **9**, 293-300.

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.

Suykens, J. A. K. (2001). Nonlinear modelling and support vector machines. *Proceeding of the Instrumentation and Measurement Technology Conference*, 287-294.

Vapnik, V. N. (1995). *The nature of statistical learning theory*, Springer, New York.

Vapnik, V. N. (1998). *Statistical learning theory*, Springer, New York.

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

Williams, C.K.I. and Seeger, M. (2001). Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*, **13**, 682-699.