Journal of the Korean Statistical Society (2007), **36**: *1, pp 175–182*

SUPPORT VECTOR MACHINE USING K-MEANS CLUSTERING

S. J. LEE[1], C. PARK[2], M. JHUN[3] AND J-Y. KOO[3]

ABSTRACT

The support vector machine has been successful in many applications because of its flexibility and high accuracy. However, when a training data set is large or imbalanced, the support vector machine may suffer from significant computational problem or loss of accuracy in predicting minority classes. We propose a modified version of the support vector machine using the K-means clustering that exploits the information in class labels during the clustering process. For large data sets, our method can save the computation time by reducing the number of data points without significant loss of accuracy. Moreover, our method can deal with imbalanced data sets effectively by alleviating the influence of dominant class.

AMS 2000 subject classifications. Primary 62H30; Secondary 68T05.
Keywords. Class imbalance, K-means clustering, support vector machine.

1. INTRODUCTION

The support vector machine (SVM) introduced by Cortes and Vapnik (1995) has been successful in a wide range of applications from various pattern recognition tasks to cancer classifications. The success of the SVM can be ascribed to its flexibility and classification accuracy. However, when faced with large or imbalanced data where the distribution of class labels in training data is severely unbalanced, the SVM may have some problems in its application.

The computation of the SVM is based on quadratic programming (QP) whose computing time depends on the number of training data. To speed up its computation without loss of classification accuracy, several methods have been proposed.

Received August 2006; accepted December 2006.

[1]Corresponding author. Department of Statistics, Seoul National University, Seoul 151-747, Korea (e-mail: seaphant@statcom.snu.ac.kr)
[2]Institute of Statistics, Korea University, Seoul 136-701, Korea
[3]Department of Statistics, Korea University, Seoul 136-701, Korea

Preprocessing by k-nearest neighborhood to select data points close to the decision boundary is one of them (Shin and Cho, 2003). We note that the class imbalance problem is more serious than the computational problem for large data sets because classification accuracy may drop significantly in predicting minority classes. Re-sampling and weight-control methods have been applied to tackle the class imbalance problem (Akbani *et al.*, 2004).

The SVM using the K-means clustering (KM-SVM) has been proposed to speed up the computing time without loss of accuracy in Wang *et al.* (2005). We propose to use the class information during the process of clustering. Our method can achieve faster computing time than the SVM by reducing the number of data points without significant loss of classification accuracy. Moreover, it can alleviate the influence of dominant class by exploiting class information during the process of clustering.

The paper is organized as follows. Section 2 reviews the SVM briefly. Section 3 describes the KM-SVM algorithm. In Section 4, we compare the performance of our method with other variants of the SVM on benchmark and simulated data sets, ensured by conclusions in Section 5.

2. SUPPORT VECTOR MACHINE

Consider a linear classification problem. Let $(x_1, y_1), \ldots, (x_N, y_N)$ be a training data set such that $x_i \in \mathcal{X} \subset R^m$ and $y_i \in \{-1, +1\}$ for $i = 1, \ldots, N$, where $\mathcal{X}$ denotes an input space. x_i and y_i are called an input and a class label, respectively. Denote the inner product and the norm in R^m as $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. The classification is performed by constructing $f(x) = \langle w, x \rangle + b$, mapping from $\mathcal{X} \to R$, such that its sign, $\text{sign}(f)$, decides the class assignment of an input $x \in \mathcal{X}$. Here w and b are called the weight vector and the bias, respectively. To obtain the solution w, the SVM solves the following quadratic optimization problem:

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i \tag{2.1}$$

$$\text{subject to } y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1, \ldots, N,$$

where $C > 0$ is the regularization parameter. Usually we solve the Wolfe dual form of the primal form (2.1):

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{N} \alpha_i \tag{2.2}$$

$$\text{subject to} \sum_{i=1}^{N} y_i \alpha_i = 0, \ \forall i : 0 \leq \alpha_i \leq C,$$

where α_i is a Lagrange multiplier corresponding to x_i. Let $\widehat{\alpha}_i$, $i = 1, \ldots, N$, be the solution of (2.2). Then the solution function is

$$\widehat{f}(x) = \sum_{i=1}^{n} \widehat{\alpha}_i y_i \langle x_i, x \rangle + \widehat{b}. \tag{2.3}$$

Since the solution can be represented in terms of non-zero $\widehat{\alpha}_i$'s alone, those nonzero $\widehat{\alpha}_i$'s are called the support vectors. $\widehat{b}$ is determined via the Karush-Kuhn-Tucker boundary conditions. Nonlinear classification can be implemented by replacing the inner product $\langle \cdot, \cdot \rangle$ by a nonlinear kernel $k(\cdot, \cdot)$. By solving the dual optimization problem in (2.2), the solution for the nonlinear classification problem has an equivalent form as (2.3) with the inner product replaced by the kernel. An example of commonly adopted kernel in nonlinear classifications is the radial basis function kernel, defined as $k(x, x') = \exp(-\gamma \|x - x'\|^2)$, where $\gamma > 0$ is a scaling parameter. For details, see Vapnik (1998) or Cristianini and Shawe-Taylor (2000).

3. K-MEANS SVM

The K-means clustering (Macqueen, 1967) is one of the most popular clustering methods. The idea of the K-means clustering is very simple. Given a data set, we assume that there is an unobserved "cluster ID" corresponding to the data set. This can be modeled through a mixture model. The K-means clustering algorithm finds a solution by maximizing the likelihood. This algorithm can deal with continuous variables only because it is based on the squared Euclidean distance. It finds local optima by minimizing the sum of the distance between each data point and its closest cluster center. The number of clusters K should be given as an input.

The support vector machine using the K-means clustering is a sequential algorithm combining the SVM with the K-means clustering. First the K-means clustering algorithm is applied to the training data. For given K, the K-means clustering yields clusters $C_1, \ldots, C_K$. For each $i = 1, \ldots, K$, the class label $\widetilde{y}_i$ for the center of C_i is determined by majority voting. Then we construct a classifier on the cluster centers using the SVM. Note that the computing time for the SVM is closely related with the number of support vectors. The application of the K-means clustering reduces the number of data points, which in turn reduces the

number of support vectors. Hence the KM-SVM has an advantage over the SVM in its computational time.

Note that the K-means clustering is an unsupervised learning algorithm that uses only the information contained in the inputs without using the information in the class labels. The method proposed in Wang *et al.* (2005). applies the K-means clustering to the whole training data directly. Let us call the method the global KM-SVM. Our method, called the by-class KM-SVM, applies the K-means clustering for each class separately. In our method, new class label for each cluster center is determined by the class to which the cluster belongs. By doing so, different data structure in different classes can be considered. We expect that our method may improve the classification accuracy of global KM-SVM by exploiting the information in the class labels while retaining the merit of the global KM-SVM, *i.e.*, saving the computing time by reducing the number of data points. Furthermore, our idea can be useful for data with class imbalance problem. This will be illustrated in Section 4.

For the KM-SVM algorithms, the number of clusters, denoted by K, is a parameter to be estimated in addition to tuning parameters for the SVM such as the regularization parameter C from (2.1) and the kernel parameter γ (Cristianini and Shawe-Taylor, 2000). As an abuse of notation, the proportion(%) denotes the number of clusters with respect to the size of original training data. The following is the algorithm of the KM-SVM's:

Step 1	Choose a set of tuning parameters (C, γ, K).
Step 2	Run the K-means clustering on training data.
Step 3	Assign new class label $\widetilde{y}_k$ for each cluster center.
Step 4	Fit the SVM on the cluster centers.
Step 5	Calculate the test error rate.
Step 6	Iterate Step 1-5 over prespecified combinations of parameters.
Step 7	Construct the optimal classifier.

The difference of the global KM-SVM and by-class KM-SVM lies in Step 2 and Step 3. In the by-class KM-SVM, the number of clusters assigned to each class is proportional to the class size in the original training data.

4. EXAMPLES

4.1. Benchmark data

We analyzed three benchmark data sets, *ionosphere*, *wdbc* and *pima-indians-diabetes* from the UCI Machine Learning Repository (http://www.ics.uci.edu/

~mlearn). They are binary classification data sets without any missing values. Table 4.1 describes characteristics of these data sets. Our experiments were carried out on 1.40Ghz Pentium PC with 768MB of main memory and our code was written in R language (http://http://www.r-project.org/). We adopted 5-fold cross validation (CV) to select the number of clusters K and the tuning parameters C and γ. Each data set was partitioned at random into training and test data with the ratio of $(2:1)$. To assess the sampling variability, the process of random partitioning were replicated 100 times. We applied the radial basis function kernel introduced in Section 2. The optimal parameters, $(\hat{C}, \hat{\gamma}, \hat{K})$, were obtained by grid search on $2^{-5}, 2^{-4}, \ldots, 2^{5}$ for C and $2^{-10}, 2^{-9}, \ldots, 2^{10}$ for γ.

TABLE 4.1 *Description of the data sets*

Data set	*No. of data*	*No. of feature*	*No. of majority/minority class*
ionosphere	351	34	225/126
wdbc	569	30	357/221
pima	768	8	500/268

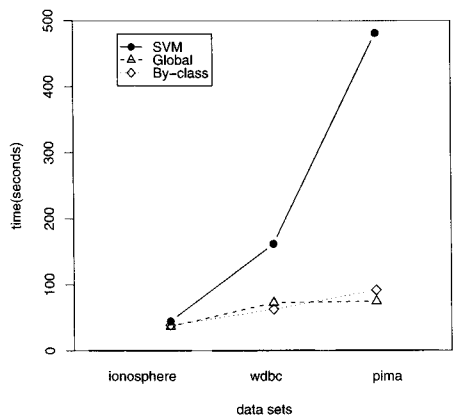

FIGURE 4.1 *Computing time in seconds for optimal models.*

Figure 4.1 shows the computing time for optimal model over 100 replications. Since the computing time is proportional to the number of data points, the KM-SVM's take much less computing time than the SVM as the size of data set increases. For example, on the largest data set, *pima* data, the global KM-SVM takes only 16% of the standard SVM fitting time. As we expected, the SVM was slightly better than KM-SVM's in terms of test error rate.

TABLE 4.2 *The mean and standard error of test error rate and the number of support vectors for UCI data sets*

Data set		*standard SVM*	*global KM-SVM*	*by-class KM-SVM*
ionosphere	*ERR*	0.060(0.002)	0.076(0.003)	0.075(0.005)
	No. of sv	105.9(3.46)	58.7(2.24)	73.2(1.93)
wdbc	*ERR*	0.026(0.001)	0.031(0.001)	0.030(0.001)
	No. of sv	62(1.84)	43.6(1.48)	39.3(1.60)
pima	*ERR*	0.23(0.0024)	0.25(0.0027)	0.24(0.0029)
	No. of sv	290.9(2.16)	132.4(5.84)	142.8(4.31)

In Table 4.2, the mean error rate of the by-class KM-SVM was almost same as that of the standard SVM and the KM-SVM's used less support vectors than the standard SVM. On *ionosphere* data, the global KM-SVM reduced 45% of support vectors with only 1.6% of loss of accuracy with respect to the standard SVM. For *wdbc* data, the reduction of support vectors by the by-class KM-SVM was about 37% while its loss in error rates was 0.4%. *Pima* data showed 51% reduction of support vectors and 1% of loss of accuracy by the by-class KM-SVM. In terms of test error, our by-class KM-SVM performed consistently better than the global KM-SVM for the benchmark data sets. Compared to the standard SVM, the by-class KM-SVM reduced the number of support vectors. However, the relative gain in computing time of the by-class KM-SVM over the global KM-SVM is not clear. As the results suggest, our method may be useful in analyzing large data sets because it reduces the computing time without significant loss of classification accuracy. In particular, our method seems to classify a bit more accurately than the global KM-SVM.

4.2. Imbalanced data

Recently, there has been increasing interest in class imbalance problem because of the increasing need for performing classification tasks in many applications with imbalanced class distribution. The areas of application include gene profiling, fraud detection, and medical diagnosis. One might consider random sampling or weighting misclassification cost according to the class size. However, these methods may not be sufficient.

In general, the SVM is believed to be more resistant to the class imbalance than other classification methods. However, it may also fail if the class distribution is severely imbalanced because the optimal hyperplane tends to lean to the side of majority class. In this subsection, we illustrate that our KM-SVM has an

advantage over other methods in the presence of class imbalance in training data through a simulation.

We consider a simple two-class imbalanced data set. Positive and negative classes are assumed to follow independent bivariate normal distributions with different mean vectors $\mu_1 = (0.5, 0)$ and $\mu_2 = (-0.5, 0)$ and common covariance matrix I. We generated random samples of size 1000 with the proportion of the majority class from 70% to 99%. The ratio of training, validation, and test data set was (3 :1 :1). The regularization parameter C was determined through grid search on $2^{-5}, 2^{-4}, \ldots, 2^{5}$. To assess the variability, this process was repeated 100 times. For imbalanced data, we applied the K-means clustering on the majority class alone. We compared the performance of our KM-SVM, standard SVM, and under-sampling SVM (US-SVM) (Akbani *et al.*, 2004). Since the Bayes decision boundary is linear, we have adopted the linear kernel.

In general classification tasks, the sensitivity, defined as the accuracy on the positive class, and the specificity, the accuracy on the negative class, have been used to evaluate the performance of a classifier. We have adopted the balanced correct-classification rate (BCR), defined as the product of the sensitivity and the specificity, as the measure of accuracy.

TABLE 4.3 *The mean and standard error of BCR*

ratio	*standard SVM*	*US-SVM*	*KM-SVM*
(99:1)	0	0.1530(0.024)	0.4261(0.031)
(95:5)	0	0.4004(0.015)	0.4940(0.011)
(90:10)	0	0.4285(0.001)	0.4723(0.008)
(80:20)	0	0.4593(0.007)	0.4783(0.005)
(70:30)	0.2140(0.019)	0.4656(0.006)	0.4755(0.005)

Table 4.3 shows the results of 100 replications for the settings of defined parameters. In terms of BCR, our KM-SVM outperformed the SVM and US-SVM. For the ratio (99:1), the performance of the US-SVM significantly dropped (0.1530), while our KM-SVM showed similar performance (0.4261). For severely imbalanced cases, the standard SVM assigned all test data to the majority class. The performance of under-sampling method was competitive in this example.

5. DISCUSSION

We introduced the KM-SVM adopting the K-means clustering. According to the assignments of class labels for cluster centers, two different versions of the KM-SVM have been considered. The results illustrate that our method can

improve the classification accuracy of the original KM-SVM by exploiting the information in class labels and it uses less support vectors than the SVM by reducing the number of data points through clustering. In this sense, our KM-SVM can be a competitive classification method for large data sets. In addition, we illustrated through a simulation that our method can also be useful for data with class imbalance problem.

A problem with our method is that our method may be sensitive to the choice of initial centers. More stable method of data reduction than the K-means clustering may be considered. Also, it would be worthwhile to investigate how much ensemble methods such as bagging or boosting can improve the predictive performance of our KM-SVM. We leave these issues as future work.

ACKNOWLEDGEMENTS

The authors are grateful to anonymous reviewers, an associate editor, and the editor for the detailed and helpful comments. The research of Park, Jhun and Koo was supported by Korea Research Foundation Grant funded by Korea Government (MOEHRD, Basic Research Promotion Fund) (KRF-2005-070-C00020). The research of Lee and Koo was supported by the SRC/ERC program of MOST/KOSEF (R11-2000-073-00000).

REFERENCES

AKBANI, R., KWEK, S. AND JAPKOWICZ, N. (2004). "Applying support vector machines to imbalanced datasets", *Proceedings of the 15th European Conference on Machine Learning*, Pisa, Italy, 39–50.

CORTES, C. AND VAPNIK, V. (1995). "Support-vector networks", *Machine Learning*, **20**, 273–297.

CRISTIANINI, N. AND SHAWE-TAYLOR, J. (2000). *An Introduction to Support Vector Machines*, Cambridge University Press.

JAPKOWICZ, N. (2000). "Learning from imbalanced data sets: a comparison of various strategies", *AAAI Workshop on Learning from Imbalanced Data Sets*, Menlo Park, CA, AAAI Press.

MACQUEEN, J. B. (1967). "Some methods for classification and analysis of multivariate observations", *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 281–297.

SHIN, H. J. AND CHO, S. (2003). "Fast pattern selection for support vector classifiers", *Proceedings of 7th Pacific-Asis Conference on Knowledge Discovery and Data Mining*, Seoul, Korea, 376–387.

VAPNIK, V. N. (1998). *Statistical Learning Theory*, Wiley-Interscience, New York.

WANG, J., WU, X. AND ZHANG, C. (2005). "Support vector machines based on K-means clustering for real-time business intelligence systems", *International Journal of Business Intelligence and Data Mining*, **1**, 54–64.