Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

SVD-LSSVM and its application in chemical pattern classification*

TAO Shao-hui, CHEN De-zhao†‡, HU Wang-ming
(*Department of Chemical Engineering, Zhejiang University, Hangzhou 310027, China*)
†E-mail: dzc@zju.edu.cn

Received May 28, 2006; revision accepted July 28, 2006

Abstract: Pattern classification is an important field in machine learning; least squares support vector machine (LSSVM) is a powerful tool for pattern classification. A new version of LSSVM, SVD-LSSVM, to save time of selecting hyper parameters for LSSVM is proposed. SVD-LSSVM is trained through singular value decomposition (SVD) of kernel matrix. Cross validation time of selecting hyper parameters can be saved because a new hyper parameter, singular value contribution rate (SVCR), replaces the penalty factor of LSSVM. Several UCI benchmarking data and the Olive classification problem were used to test SVD-LSSVM. The result showed that SVD-LSSVM has good performance in classification and saves time for cross validation.

Key words: Pattern classification, Structural risk minimization, Least squares support vector machine (LSSVM), Hyper parameter selection, Cross validation, Singular value decomposition (SVD)
doi:10.1631/jzus.2006.A1942 **Document code:** A **CLC number:** TP183

INTRODUCTION

Pattern classification is an important problem in the machine learning field and least squares support vector machine (LSSVM) proposed by Suykens is an easy and powerful tool for this problem (Suykens and Vandewalle, 1999a). Only a set of linear equations should be solved during training of an LSSVM, which makes it easy to be realized. LSSVM is based on structural risk minimization (SRM) rule, which enhances its generalization ability. SRM rule requires that model complexity and empirical risk of the model must be minimized at the same time and balanced well. To carry SRM well, proper hyper parameters such as kernel function parameter and penalty factor must be selected carefully. The most popular way to select the hyper parameters for LSSVM is the *f*-fold cross-validation. The computation load of cross validation increases as the size of training sample increases, while the classifiers are usually used to process massive data in data mining field, so it will be a very long time procedure to choose hyper parameters through cross validation. To save cross validation time for LSSVM with radial basis function (RBF) as kernel function, LSSVM trained with the singular value decomposition of the kernel matrix, i.e. SVD-LSSVM, is proposed in this paper. The new version of LSSVM carries SRM rule in a different way from original LSSVM.

LSSVM BASED ON SINGULAR VALUE DECOMPOSITION OF KERNEL MATRIX

Simplified version of LSSVM

To reduce computation complexity of training LSSVM, sometimes the intercept term b in the decision function can be canceled if the dependent variable y is centered to zero mean (Pelckmans *et al*., 2005; 2006). The constraint optimization problem of binary LSSVM without intercept term can be written as follows:

‡ Corresponding author
* Project (No. 20276063) supported by the National Natural Science Foundation of China

$$\min J(\boldsymbol{w}, e_i) = \frac{1}{2}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \gamma \sum_{i=1}^{n} e_i^2 \right), \quad \text{s.t. } \boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x}_i) = y_i - e_i, \ i = 1, 2, \ldots, n, \tag{1}$$

here $\boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x}_i)$ is the linear classifier in the feature space, e_i is the error on the ith training example and $\sum_{i=1}^{n} e_i^2$ is the empirical risk, γ is the LSSVM's penalty factor. As γ increases, the empirical risk will decrease while the model complexity $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$ will increase.

To solve the constrained optimization problem Eq.(1), the linear equations set Eq.(2) for finding the Lagrange multipliers must be set up (Pelckmans *et al*., 2005; 2006):

$$[\boldsymbol{K}+\gamma^{-1}\boldsymbol{I}]\,\boldsymbol{\alpha}=\boldsymbol{y}, \tag{2}$$

here $\boldsymbol{K}$ is the kernel matrix and the ith row and the jth column element of $\boldsymbol{K}$ is $k_{ij}=k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. After α_i is solved, the class of $\boldsymbol{x}$ is judged by the sign of $y=\sum_{i=1}^{n} k(\boldsymbol{x}, \boldsymbol{x}_i)\alpha_i$.

According to the matrix computation theory, the computation complexity of solving one $n\times n$ real symmetry linear equations is $O(n^3)$ when it is solved by Cholesky decomposition (Golub and van Loan, 1989). Of course the simplified LSSVM can also be applied directly to multi classification problem, and the training computation complexity is $l\times O(n^3)$, where l is the dimension of the class label vector (Suykens and Vandewalle, 1999b).

Carrying SRM for LSSVM through SVD of kernel matrix

In Eq.(1), the penalty factor γ balances the model complexity $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$ and the empirical risk during the optimization, while the equations set Eq.(2) shows that the balance is realized in a way similar to ridge regression. Now a new way to realize the balance through SVD of the kernel matrix will be proposed in the following, which can save cross validation time for selecting hyper parameters.

Supposing SVD of $\boldsymbol{K}$ is written as Eq.(3):

$$\boldsymbol{K} = \sum_{i=1}^{n} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^{\mathrm{T}}, \ \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n-1} \geq \lambda_n, \tag{3}$$

here λ_i is the singular value. If the kernel is RBF function with width parameter σ, $\boldsymbol{K}$ is a definite symmetry matrix and $\lambda_i\geq 0$ ($1\leq i\leq n$). If only first p singular values of $\boldsymbol{K}$'s SVD are kept, i.e. $\boldsymbol{K}=\sum_{i=1}^{p} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^{\mathrm{T}}$, $1\leq p\leq n$, the solution of $\boldsymbol{K}\cdot\boldsymbol{\alpha}=\boldsymbol{y}$ can be written as Eq.(4):

$$\boldsymbol{\alpha}(p) = \sum_{i=1}^{p} [(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{y}/\lambda_i)\cdot\boldsymbol{u}_i] = \sum_{i=1}^{p} [(z_i/\lambda_i)\cdot\boldsymbol{u}_i]. \tag{4}$$

When the solution is $\boldsymbol{\alpha}(p)$, the sum squared remains of $\boldsymbol{K}\cdot\boldsymbol{\alpha}=\boldsymbol{y}$, which equals the empirical risk in Eq.(1), can be represented by Eq.(5) (Golub and van Loan, 1989):

$$\sum_{i=1}^{n} e_i^2 = \| \boldsymbol{K}\boldsymbol{\alpha}(p) - \boldsymbol{y} \|_2^2 = \sum_{i=p+1}^{n} (\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{y})^2. \tag{5}$$

And the $\boldsymbol{w}(p)$ can be written as $\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\alpha}(p)$, it is easy to prove Eq.(6), where $z_i=[\boldsymbol{\alpha}(p)]^{\mathrm{T}}\boldsymbol{u}_i$.

$$\| \boldsymbol{w}(p) \|_2^2 = [\boldsymbol{\alpha}(p)]^{\mathrm{T}} \boldsymbol{K}\boldsymbol{\alpha}(p) = \sum_{i=1}^{p} (z_i^2/\lambda_i). \tag{6}$$

So as p increases, the empirical risk $\sum_{i=1}^{n} e_i^2$ of Eq.(1) will decrease while the model complexity $[\boldsymbol{w}(p)]^{\mathrm{T}}\cdot\boldsymbol{w}(p)$ in Eq.(1) will increase. When $p=n$, the empirical risk will be 0 while $[\boldsymbol{w}(p)]^{\mathrm{T}}\cdot\boldsymbol{w}(p)$ will have the maximum value. And a proper p can be selected to balance the empirical risk and the structural risk.

Set a new hyper parameter called singular value contribution rate (SVCR), which is defined as Eq.(7):

$$\eta = \sum_{i=1}^{p} \lambda_i \Big/ \sum_{i=1}^{n} \lambda_i. \tag{7}$$

It is easy to know that when p increases from 0 to n, η will increase from 0 to 100%. So the selection of p can be realized by selecting η. Then the SVD-LSSVM algorithm can be set up. It is very easy to generalize this algorithm to multi class problem (Suykens and Vandewalle, 1999b).

In (Pelckmans *et al*., 2005), SVD is also used to train LSSVM whose linear equations set is subjected to a nonlinear constraint restricting the empirical risk to be equal to the noise level estimated via differor-

gram method. But all the singular values are used there, which is different from SVD-LSSVM.

To show how the SVCR influences SVD-LSSVM's performance on the training and test data, the kernel parameter σ was fixed as 1.5 while the SVCR increased, then the leave-50-out cross validation method was used on the Breast problem of UCI benchmarking data (Blake and Merz, 1998) to observe how the fitting and prediction abilities varied as the SVCR increased. Fig.1 shows that the fitting error decreases as the SVCR increases; the prediction error also decreases at the early time of SVCR's increasing but once the SVCR exceeds a certain value, the prediction error increases.

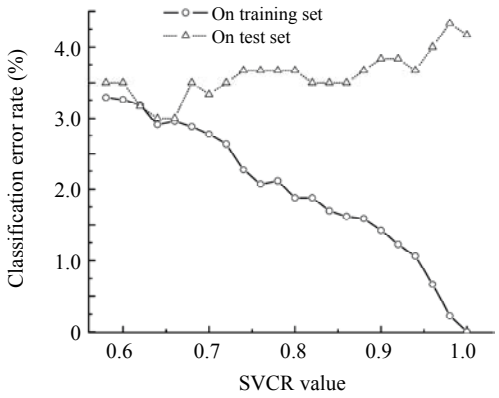

Fig.1 The influence of SVCR value on the training and prediction classification error rates

It can be seen that LSSVM balances the model complexity and the empirical risk of the machine by introducing γ while SVD-LSSVM balances them by choosing a proper SVCR value η, which normally will be chosen in [0.6, 1]. A virtue of SVD-LSSVM is that it will save time for choosing hyper parameters, which will be discussed in the following section.

Computation complexity of choosing hyper parameters

In this paper the hyper parameters of LSSVM and SVD-LSSVM are both selected by f-fold cross validation. The hyper parameters combination that gives the least mean of f classification error rate (*CER*) on the validation set is taken as the hyper parameters of the machine. *CER* for a dataset is defined as $CER=(n_{\text{error}}/n_{\text{classified}})\times 100\%$, here $n_{\text{classified}}$ is the size of the entire classified dataset and n_{error} is the number of examples that is misclassified or failed to classify.

For a simplified LSSVM with l-element class label vector, supposing there are c candidates for kernel function parameter σ, g candidates for the penalty factor γ and the training/validation set is partitioned into f groups. To select hyper parameters, $f\times c\times g$ LSSVM training must be carried. So $l\times f\times c\times g$ linear equations set like Eq.(2) must be solved and the computation complexity of the f-fold cross validation is $l\times f\times c\times g\times O(n^3)$.

While for SVD-LSSVM with an l-element class label vector, supposing there are c candidates for kernel function parameter σ, s candidates for SVCR η. Once σ is determined, the SVD of $\boldsymbol{K}$ like Eq.(3) is fixed and it has nothing to do with η value. So only $f\times c$ SVD of $\boldsymbol{K}$ need to be carried for the f-fold cross validation, even for multi class problem, because for all the linear equations sets of multi class LSSVM they have the same kernel matrix. Because the computation complexity of SVD of $n\times n$ matrix is $O(n^3)$ (Golub and van Loan, 1989), the main computation load of SVD-LSSVM cross validation is $f\times c\times O(n^3)$.

Although computation load of carrying matrix SVD is a little larger than solving a linear equations set whose coefficient matrix has the same size (Golub and van Loan, 1989), the number of linear equations set to be solved is $l\times g$ times of the SVD of matrix if the number of σ candidates is the same for LSSVM and SVD-LSSVM. So the computation complexity of the hyper parameter selection can be decreased using SVD-LSSVM, which will be shown in the next section.

TEST AND APPLICATIONS

Description of the tested datasets

To evaluate the classification performance of SVD-LSSVM, four UCI benchmarking data, i.e. Heart, Breast, Ionosphere and Wine problem (Blake and Merz, 1998) and one chemical pattern classification problem, i.e. Olive problem (Hopke and Massart, 1993) were tested. The basic information on these datasets is listed in Table 1.

When comparing classification performance of different classifiers, the data must be partitioned as training set and test set and for all the problems. In this paper about 2/3 of the entire data are taken as the

Table 1 The tested dataset and partitioning of training and test data

Data set	Class number	Dimension of input vector	Size of training/validation set	Size of test set
Heart	2	13	190	107
Ionosphere	2	9	230	121
Breast	2	34	450	283
Wine	3	13	120	58
Olive	9	8	380	192

training/validation set and the other 1/3 are taken as the test set. The detailed partitioning of training and test set for each problem is listed in Table 1.

Another important problem is the preprocessing of data. In this paper the only preprocessing is to normalize all the examples' input to zero mean and unit standard deviation; the examples' class label vector are normalized to zero mean.

Hyper parameters selection

To evaluate the classification performance of SVD-LSSVM, simplified LSSVM, standard SVM, Bayesian classifier and multi feed-forward neural networks (MFNN) with single hidden layer were also applied to these problems for comparison purpose. The number of MFNN hidden layer neurons was determined by experience. In this paper, there are 4 hidden neurons for Heart problem, 5 hidden neurons for Ionosphere, 6 for Breast, 2 for Wine and 9 for the Olive problem. The hyper parameters of SVD-LSSVM, SVM and the simplified LSSVM were selected via the 10-fold cross validation. The candidate set of the kernel parameter σ for SVM, LSSVM and SVD-LSSVM is {0.5, 5, 10, 15, 25, 50, 100, 250, 500}$\cdot m^{0.5}$, where m is the dimension of the input vector. The penalty factor γ for LSSVM and SVM was selected from {0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000} (van Gestel *et al*., 2004), while η for SVD-LSSVM was selected from {0.8, 0.82, 0.84, 0.86, 0.88, 0.90, 0.92, 0.94, 0.96, 0.98, 1}. Ten-fold cross validation was used to select hyper parameters for all the test problems. So f=10, c=9, g=11 and s=11.

All the experiments were carried out on a personal computer with AMD Athlon 1.5 GHz CPU and 512 M memory, programming language Matlab 7 for Windows 2000. The time needed for hyper parameters selection of SVD-LSSVM and the simplified LSSVM via cross validation for each problem are listed in Table 2, where CV represents the time measured in seconds spent for selecting hyper parameters using ten-fold cross validation method for LSSVM and SVD-LSSVM. It can be seen that for all the problems SVD-LSSVM spends less time than the simplified LSSVM.

Table 2 Comparing cross validation time for LSSVM and SVD-LSSVM

Data set	CV (s)	
	LSSVM	SVD-LSSVM
Heart	29	11
Ionosphere	107	30
Breast	184	139
Wine	12	5
Olive	136	113

Since carrying out SVD of a matrix is an iteration procedure, the time spent for SVD varies with the value of matrix elements. So how much cross validation time can be saved by SVD-LSSVM is connected not only with the kernel matrix size but also with the value of the kernel matrix elements. According to Table 2, SVD-LSSVM spends less time than LSSVM for all the problems especially for Ionosphere problem. While for Olive problem the saved time is the least. This may be caused by the difference of the data.

For the same problem but different training/validation set the values of hyper parameters selected by cross validation will be different. The reason is probably that the optimal values of hyper parameters depend on the training/validation dataset. Table 3 lists the optimal hyper parameters for LSSVM and SVD-LSSVM that is selected for each problem for one random experiment. Obviously their kernel parameters are the same except for the Ionosphere problem.

Table 3 Hyper parameters selected for LSSVM and SVD-LSSVM

Dataset	LSSVM		SVD-LSSVM	
	σ	γ	σ	η
Heart	36.0555	1	36.0555	0.9
Ionosphere	2.8723	5	28.7230	0.8
Breast	1.5000	1	1.5000	0.8
Wine	18.0280	10	18.0280	0.8
Olive	1.4142	10	1.4142	0.9

Comparison of classification performance

Random experiment was used to evaluate the performance of the five classifiers. For each experiment the entire examples of the dataset were randomly sorted and then the first 2/3 examples were taken as training/validation set and the other 1/3 were taken as the test set. The hyper parameters were selected by the *f*-fold cross validation on the training/validation set. Once the hyper parameters were determined, then the classifier was trained with the entire training/validation set with the selected hyper parameters. After the classifier was trained, a *CER* on the training set, which is noted as CER_{fit}, and a *CER* on the test set, which is noted as CER_{test}, can be observed. To avoid chanciness when evaluating, 20 such random experiments were carried out for each problem. The mean and deviation of the 20 CER_{fit} and 20 CER_{test} were taken as the scales of learning ability (fitting ability) and the generalization ability (prediction ability) respectively. Their standard deviations can scale the stability of the classifier.

Table 4 lists the mean and standard deviations of CER_{fit} and CER_{test} for all the five classifiers, i.e. Bayesian classifier, MFNN, SVM, simplified LSSVM and SVD-LSSVM on all the benchmarking datasets for the 20 random experiments.

Table 4 shows that MFNN classifiers have the worst performance on test set compared with all the other classifiers. The reason may be the MFNN structure is hard to be tuned and a global optimized solution cannot always be found in training MFNN. This will decrease the generalization ability of MFNN. The Bayesian classifier is the most simple and fast but its classification performance depends on the statistical distribution of the training data, so except Heart and Wine problems, it performs worse than SVM classifiers.

The performance of SVM, LSSVM and SVD-LSSVM is similar for 3 binary classification problems. But for the multi classification problems, i.e. Wine and Olive, SVM performs a little better than the two LSSVM versions, because several SVM classifiers are combined to solve the multi classification problem while for LSSVM and SVD-LSSVM only one classifier is used. SVD-LSSVM's performance is comparable to LSSVM, while it has saved time for cross validation.

CONCLUSION

This paper proposed a new version of LSSVM trained through singular value decomposition of the kernel matrix, which is called SVD-LSSVM here. The new version of LSSVM presented a new way to carry out SRM rule. Its empirical risk and model complexity are balanced by a new hyper parameter, i.e. singular value contribution rate η whose chosen scope is [0.6, 1] for all the classification data. In this paper, hyper parameters of LSSVM and SVD-LSSVM are selected via *f*-fold cross validation. SVD of the kernel matrix only relates to the kernel parameter while it has nothing to do with SVCR value η and the class label vector's dimension, and this saves time for selecting SVD-LSSVM hyper parameters compared to LSSVM. The result of applying this new version of LSSVM to four benchmarking classification datasets and one chemical pattern classification problem showed that SVD-LSSVM could save much time for hyper parameter selection by cross validation

Table 4 Performance comparison of the different classifiers

Data		Bayes	MFNN	SVM	LSSVM	SVD-LSSVM
Breast	CER_{fit} (%)	3.67±0.70	2.21±0.38	2.33±0.39	1.50±1.25	1.94±0.93
	CER_{test} (%)	4.16±1.30	3.22±2.84	3.15±0.15	3.76±1.23	3.26±0.87
Heart	CER_{fit} (%)	13.92±1.38	13.00±2.14	14.03±2.18	12.87±2.65	14.47±1.70
	CER_{test} (%)	17.10±2.54	19.81±2.95	18.55±3.24	16.96±2.67	17.14±2.83
Ionosphere	CER_{fit} (%)	8.43±1.27	5.67±3.66	0.91±0.70	0.20±0.33	4.50±1.47
	CER_{test} (%)	13.26±2.91	13.93±3.62	9.46±2.84	6.36±2.04	6.20±2.35
Wine	CER_{fit} (%)	0.08±0.26	0.46±0.79	0.33±0.57	0.21±0.66	1.13±1.36
	CER_{test} (%)	1.12±1.16	4.91±3.97	2.84±1.61	3.53±2.40	3.79±2.54
Olive	CER_{fit} (%)	5.78±0.85	5.20±1.00	0.85±0.91	0.55±0.69	1.26±0.65
	CER_{test} (%)	6.85±1.69	11.64±2.99	4.95±2.02	5.13±1.31	5.03±1.40

while its classification performance does not decrease compared with original LSSVM.

References

Blake, C.L., Merz, C.J., 1998. UCI Repository of Machine Learning Database. Http://www.ics.uci.edu/~mlearn/mlrepository.html. Dept. of Information and Computer Science, University of California, Irvine, CA.

Golub, G.H., van Loan, C.F., 1989. Matrix Computations. Gene Johns Hopkins University Press.

Hopke, P.K., Massart, D.L., 1993. Reference data sets for chemometrical methods testing. *Chemometrics and Intelligent Laboratory Systems*, **19**(1):35-41. [doi:10.1016/0169-7439(93)80080-2]

Pelckmans, K., de Brabanter, J., Suykens, J.A.K., de Moor, B., 2005. The differogram: Nonparametric noise variance estimation and its use for model. *Neurocomputing, Special Issue on Signal Processing*, **69**(1-3):100-122.

Pelckmans, K., Suykens, J.A.K., de Moor, B., 2006. Additive regularization trade-off: Fusion of training and validation levels in kernel methods. *Machine Learning*, **62**(3):217-252. [doi:10.1007/s10994-005-5315-x]

Suykens, J.A.K., Vandewalle, J., 1999a. Least squares support vector machine classifiers. *Neural Processing Letters*, **9**(3):293-300. [doi:10.1023/A:1018628609742]

Suykens, J.A.K., Vandewalle, J., 1999b. Multiclass Least Squares Support Vector Machines. Intl. Joint Conference on Neural Networks, IJCNN'99, Washington, D.C.

van Gestel, T., Suykens, J.A.K., Baesens, B., Stijn, V., Vanthienen, J., Dedene, G., Bart, D.M., Vandewalle, J., 2004. Benchmarking least squares support vector machine classifiers. *Machine Learning*, **54**(1):5-32. [doi:10.1023/B:MACH.0000008082.80494.e0]