Appl. Math. J. Chinese Univ. Ser. B
2003,18(3):332-342

SVM CLASSIFICATION:ITS CONTENTS AND CHALLENGES

Yue Shihong   Li Ping   Hao Peiyi*

Abstract. SVM (support vector machines) have become an increasingly popular tool for machine learning tasks involving classification, regression or novelty detection. In particular, they exhibit good generalization performance on many real issues and the approach is properly motivated theoretically. There are relatively a few free parameters to adjust and the architecture of the learning machine does not need to be found by experimentation. In this paper, survey of the key contents on this subject, focusing on the most well-known models based on kernel substitution, namely SVM, as well as the activated fields at present and the development tendency, is presented.

§ 1 Introduction

If you knock the word "SVM" in the SCI index tool on International network, you would take on thousands of records immediately. This shows its great effects on our world. SVM, namely, support vector machines have been successfully applied to a number of applications ranging from particle identification and text categorization to engine knock detection, bioinformatics and database marketing[1-6]. The approach is systematic and properly motivated by statistical learning theory[7]. Training involves the optimization of a convex cost function that there are no false local minima to complicate the learning process. This approach has many other benefits. For example, the model constructed has an explicit dependence on the most informative patterns in data(support vectors). Hence the interpretation is straightforward and the data cleaning[8-10] could be implemented to improve performance. SVM are the most effective class of algorithms that use the idea of kernel substitution which we will broadly refer to a kernel method. Although SVM are researched deeply abroad and have obtained great success, however, few of people understand it in our country.

The present review involves the main ideas of SVM algorithms for clustering and

Received:2002-04-12.

MR Subject Classification:03C43.

Keywords:kernel methods, mathematical programming, SVM.

Supported by the National 863 Plan Foundation of China (2002AA412010).

reports the applications based on optimal character recognition. We do not attempt a full treatment of all available literature and fields, but we present a somewhat biased point of view illustrating the main ideas drawn mainly from the work of many authors and providing the best of our knowledge reference to related works for further reading. We hope that it nevertheless will be useful for the readers. It differs from other reviews, such as the ones in literatures[11-13], mainly in the choice of the presented materials, that are all-embracing whereas rough. We place more emphasis on the latest researching tendency, or challenge for users, and on connection to key techniques of SVM.

We start by presenting some basic concepts and techniques of SVMs in § 1. Then we introduce the idea of *kernel feature space* and the original SVM approach, its implementation and some variants. The important properties of SVM will be devoted to question of model in § 2. Subsequently, we discuss the major difficulties of *kernel-based methods* learning in § 3. Current developments and open issues are remarked in § 4. Finally, we sum up our several view points in § 5.

§ 2 What is a support vector machine(SVM)classifier?

According to the theory of SVM, while traditional techniques for pattern recognition are based on minimization of the *empirical risk*, that is, on attempt to optimize performances on the training set, SVM minimize the *structural risk*, the probability of misclassifying yet-to-be-seen patterns for a fixed but unknown probability distribution of data. This new induction principle which is equivalent to minimizing an upper bound of generalization error relies on the theory of uniform convergence in probability.

Support vector machines perform pattern recognition between two point classes by finding a decision surface determined by certain points of the training set, termed *support vectors*(SV). This surface which in some feature space of possibility infinite dimension can be regarded as a hyperplane, is obtained from the solution of a problem of quadratic programming[14-16]. The basic SVM task is to estimate a classification function $f:\mathbf{R}^N \to \{\pm 1\}$ using input-output training data from two classes: $(x_1, y_1), \ldots, (x_l, y_l) \in \mathbf{R}^N \times \{\pm 1\}$.

First, we limit the discussion to linear classification functions. The goal is to establish the equation of a hyperplane that divides the training data leaving all points of the same class on the same side while maximizing the minimum distance between either of the two classes and the hyperplane. If points are linearly separable, then there exist an n-vector w and a scalar b such that

$$wx_i - b \geqslant 1, \text{if } y_i = 1, \text{and } wx_i - b \geqslant -1, \text{if } y_i = -1 \ i = 1, \ldots, l \tag{1}$$

or equivalently,

$$y_i(wx_i - b) \geqslant 1 \quad i = 1, \ldots, l. \tag{2}$$

The "optimal" hyperplane $wx - b = 0$ is geometrically equivalent to maximizing the maximum "margin", the distance between two parallel planes $wx-b=1$ and $wx-b=-1$ (see Fig. 1).

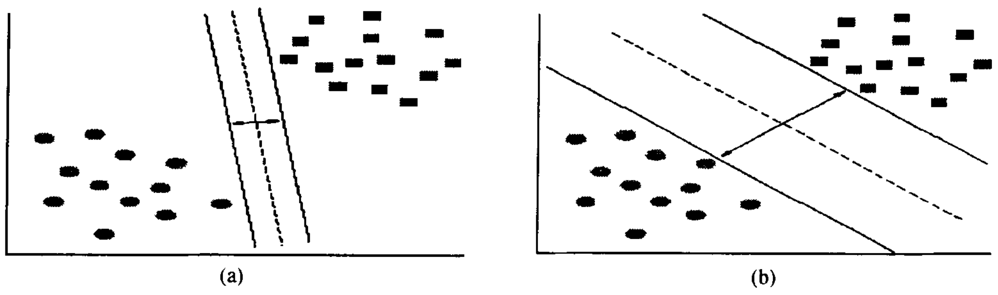

Fig. 1 The optimal plane maximizes the margin

The "margin" in Euclidean distance is $2/\|w\|_2$, where $\|w\|_2=(\sum_{i=1}^{n}w_i^2)^{\frac{1}{2}}$ is 2-norm. To maximize the margin is equivalent to minimizing 2-norm $\|w\|_2$ subject to constraint (2).

1) Structural risk minimization: According to the structural risk minimization, a typical uniform Vapnik and Chervonenkis bound which holds with probability $1-\eta$, has the following form:

$$R(\lambda)\leqslant R_{emp}(\lambda)+\sqrt{\frac{h(\ln 2lh^{-1}+1)-\ln\eta/4}{l}},\forall\ \lambda\in F, \tag{3}$$

where h is the VC-dimension of the classifier, l is the number of the training data points. In order to achieve a small structural risk, that is, the good generalization performances, both the empirical risk and the ratio between VC-dimension and the number of data points have to be small-even in an infinite-dimension; F is function class that the estimate λ chosen from. Vapnik[13] showed that, if we assume all the points $x_1,\ldots,x_l$ lie in the N-dimension sphere of radius R, the set of function: $\{f_{w,b}=\mathrm{sign}(w\cdot x+b)\,|\,\|w\|\leqslant A\}$ has a VC-dimension h that satisfies the following bound: $h\leqslant A^2R^2+1$, where R is radius of the smallest ball around the data.

The goal of SVM is to find, among the canonical hyperplanes that correctly classify data, the one with minimum norm w, because keeping this norm small will keep the VC-dimension small too, and keeping the margin larger as shown before should lead to a better generalization and prevent fitting over in high-dimensional attribute spaces[17].

2) Training support vector machine: In general, the classes will not be separable, so the generalized optimal plane (GOP) problem P1 is used. A set of variables $\Xi=\{\xi_i\}_i^d$ that measure the amount of variation of constraints is added for each point. The final GOP formulation is

$$\text{P1}\quad \min_{w,b,\xi}\frac{1}{2}\|w\|^2+C\Big(\sum_{i=1}^{l}\xi_i\Big)$$

$$\text{s. t. } y_i(wx_i + b) + \xi_i \geqslant 1, \xi_i \geqslant 0, i = 1, \ldots, l, \tag{4}$$

where $C > 0$ is a fixed penalty parameter chosen by the user, a larger C corresponds to assigning a higher penalty to errors. The parameter C can be regarded as a regularization parameter. SVM tend to maximize the minimum distance $1/w$ for small C, and minimize the number of misclassified points for large C.

Problem P1 can be solved by means of classical method of Lagrange multipliers. If we let $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_l)$ and $\Gamma = (r_1, r_2, \ldots, r_l)$ be the l nonnegative Lagrange multipliers associated with constraint of P1, the solution to problem P1 is equivalent to determining the saddle point of function

$$L(w, b, \Xi, \Lambda, \Gamma) = \frac{1}{2} \| w \|^2 + C\Big(\sum_{i=1}^{l} \xi_i \Big) - \sum \lambda_i [y_i(wx_i + b) - 1 + \zeta_i] - \sum_{i=1}^{l} r_i \zeta_i \tag{5}$$

which has to minimize (5) with respect to w, b and maximize with respect to $\Lambda \geqslant 0, \Gamma \geqslant 0$. Differentiating (5) and setting the results equal to zero, we obtain

$$\frac{\partial L(w, b, \Xi, \Lambda, \Gamma)}{\partial w} = \frac{\partial L(w, b, \Xi, \Lambda, \Gamma)}{\partial b} = \frac{\partial L(w, b, \Xi, \Lambda, \Gamma)}{\partial \xi_i} = 0, \tag{6}$$

$$\Big(w - \sum_{i=1}^{l} \lambda_i y_i x_i \Big) = \sum_{i=1}^{l} \lambda_i y_i = C - \lambda_i - r_i = 0, \tag{7}$$

$$w^* = \sum_{i=1}^{l} \lambda_i^* y_i x_i.$$

Substituting (6) (7) into (5), we have the problem P1 reduce to maximizing function

$$L(\Lambda) = \sum_{i=1}^{l} \lambda_i - \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j x_i x_j \tag{8}$$

subject to constraint (4) and $\Lambda \geqslant 0, \Gamma \geqslant 0$. This new problem is called a dual problem and can be formulated as

$$\text{P2} \quad \max_{\Lambda} \Lambda l - \frac{1}{2} \Lambda D \Lambda, \text{ s. t. } \Lambda y = 0, 0 \leqslant \Lambda \leqslant C, \tag{9}$$

where $y = (y_1, \ldots, y_1)$ and D is a symmetric $l \times l$ matrix with elements $D_{ij} = y_i y_j x_i \cdot x_j$. As for the pair (w^*, b^*), it is easy to find that $w^* = \sum_{i=1}^{l} \lambda_i^* y_i x_i$, and while b^* can be determined from Kuhn-Tucker conditions

$$\lambda_i^* (y_i(w^* x_i + b^*) - 1 + \xi_i^*) = 0 \text{ and } (C - \lambda_i^*) \xi_i^* = 0, \tag{10}$$

these points x_i for which $\lambda_i^* > 0$ are termed support vectors and close to the decision boundary.

Particularly, one would expect these points closer to the boundary of the classes more important in solution than the data points that are far away, since the former are hard to be classified. These data points, in some sense, help shape and define better the decision surface than other points. Therefore, the support vectors form a geometrical view point of border points. A direct consequence of the proceeding argument delivers another important geometrical viewpoint and algorithmic property, that, usually, support vectors are very few.

Using Eqs. (8)(9) and (10), the decision function can be written as

$$f(x) = \mathrm{sign}(wx + b^*) = \mathrm{sign}\Big(\sum_{i=1}^{l} y_i\lambda_i^*(xx_i) + b^*\Big). \tag{11}$$

3)Nonlinear decision surfaces: In a nonlinear case, the extension to more complex decision surfaces is conceptually quite simple, and is done by mapping input variable x into a higher dimensional (may be infinite)"*feature space*", and by working with linear classification in that space (see Fig. 2)

$$x \to \phi(x) = (a_1\phi(x), a_2\phi_2(x), \ldots, a_n\phi_n(x), \ldots). \tag{12}$$

Under mapping (12) the solution of SVM has the form

$$f(x) = \mathrm{sign}(\phi(x)w^* + b^*) \Rightarrow \mathrm{sign}\Big(\sum_{i=1}^{l} y_i\lambda_i^*\phi(x)\phi(x_i) + b^*\Big). \tag{13}$$

It is convenient to introduce the so-called kernel function K,

$$K(x,y) \equiv \phi(x)\phi(y) = \sum_{n=1}^{\infty} a_n^2\phi_n(x)\phi_n(y). \tag{14}$$

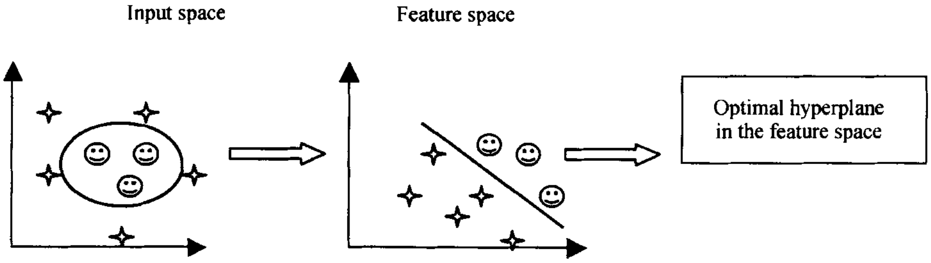

Fig. 2 Mapping the training data nonlinearly into a higher-dimensional feature space via ϕ and constructing a separate hyperplane with maximum margin

There are some commonly used kernels in Table 1.

Table 1 Some common kernel functions and the type of decision surface they define

Kernel function	$K(x,y)=\exp(-\lVert x-y\rVert^2)$	$K(x,y)=\tanh(x\cdot y-\theta)$	$K(x,y)=(1+x\cdot y)^d$
Type classifier	Gaussian RBF	Multi-layer perceptron	Polynomial of degree

Now the architecture of SVM methods can be intuitively illustrated in Fig. 3 below, where $v_i, i=1,2,\ldots$ stands for a group of coefficients of kernels.

The input x and support vectors x_i (in this example: digits) are nonlinearly mapped (by Φ) into a feature space F, where dot products are computed using kernel $k(\cdot,\cdot)$, these two layers are in practice computed in one single step. The results are linearly combined by weights v_i found by solving a quadratic program or an eigenvalue problem. The linear combination is fed into function $f(x)$ (in pattern recognition, $f(x)=\mathrm{sign}\ (x+$

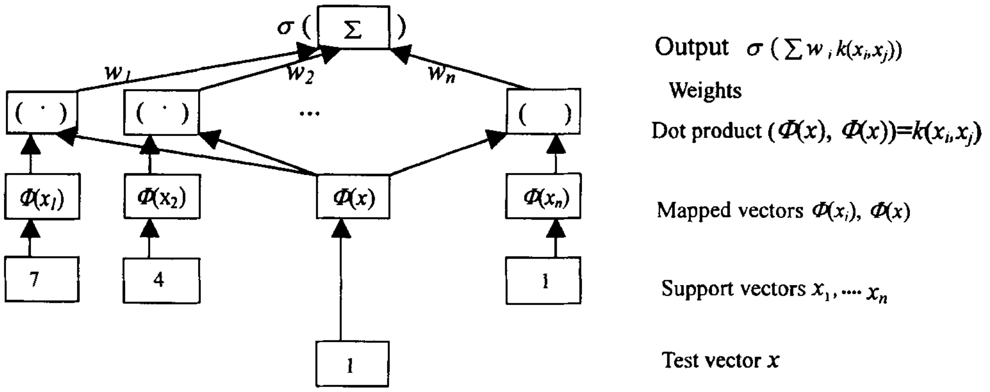

Fig. 3 Architecture of SVM methods

b);in regression estimation $f(x)=x+b$).

4)Multi-class classification[18-21]: we use the "one-against-one" approach (Knerr,et al. , 1990) classifiers are constructed each one from two deferent classes. This strategy on SVM was first used by Friedman(1996) and Knerr,et al (1999). For the training data from i-th and j-th classes, we solve thd following binary classification problem

$$\min_{w^{ij},b^{ij},\xi^{ij}} \frac{1}{2}(w^{ij})^{\mathrm{T}}w^{ij} + c\Big(\sum_t(\xi^{ij})_t\Big),$$

$$((w^{ij})^{\mathrm{T}}\Phi(x_t)) + b^{ij} \geqslant 1 - \xi_t^{ij}, \text{if } x_t \text{ in } i \text{ th class},$$

$$((w^{ij})^{\mathrm{T}}\Phi(x_t)) + b^{ij} \leqslant -1 - \xi_t^{ij}, \text{if } x_t \text{ in } i \text{ th class } \xi_t^{ij} \geqslant 0. \tag{15}$$

In classification we use a voting strategy:each binary classification is considered to be a voting where votes can be cast for all data points x, and the end point is designated to be in a class with maximum number of votes. In the case that two classes have identical votes, though it may not be a good strategy, we simply select the one with the smaller index.

Another method for multi-class classification is the "one-against-all" approach in which k SVM models are constructed and i-th SVM is trained with the examples in i-th class with positive labels and all other examples with negative labels. We do not agree with some research works(e. g. Weston and Watkins, 1998; Platt, et al. , 2000) that it does not perform as good as "one-against-one" method. In addition, though we have to train as many as classifiers since each problem is smaller (only data from two classes), the total training time may not be more than that for the "one-against-all" method. This is one of reasons why we choose the "one-against-one" method. Some detailed comparisons are shown in [17].

§ 3 Properties of SVM[22-25]

1) Decision structure of SVM: In the above, ones may see the ingenious technique of

SVM, by which a major obstacle is got over: every (linear) algorithm that only uses scalar products can implicitly be executed in feature space by using kernels[25], i. e. one can very elegantly construct a nonlinear version of a linear algorithm, furthermore, as seen in the last section, most optimization methods are based on the second-order optimality conditions, so called Karush-Kuhn-Tucker conditions which state the necessary and in some cases sufficient conditions for a set of variables to be optimal for an optimization problem. It comes handy that these conditions are particularly simple for the dual SVM problem[25],

$$\lambda_i = 0 \Rightarrow y_i f(x_i) \geqslant 1, \xi_i = 0, \lambda_i = C \Rightarrow y_i f(x_i) \leqslant 1,$$
$$\xi_i \geqslant 0 \quad 0 < \lambda_i < C \Rightarrow y_i f(x_i) = 1, \xi_i = 0. \tag{16}$$

They reveal one of the most important properties of SVM: the solution is sparse in Eq. (16), i. e. many patterns are outside the margin area and the optimal λ_i's are zero. Specifically, the KKT conditions show that only such connected to a training pattern which is either on the margin (i. e. $0<\lambda_i<C$ and $y_i f(x_i)=1$) or inside the margin area (i. e. , $\lambda_i = C$ and $y_i f(x_i) < 1$) are nonzero. Without this scare property, SVM learning would hardly be practical for large data sets. In general, the following property guarantees that SVM attain their objection properly.

2) Uniqueness of the SVM solution[22]: Some necessary and sufficient conditions for uniqueness of the support vector solution have been given for the problems of pattern recognition and estimation for a former class of cost functions. It is well-known that the standard form of SVM is defined as

$$\min_{w^{ij}, b^{ij}, \xi^{ij}, \xi^{ij*}} \frac{1}{2}(w^{ij})^{\mathrm{T}} w^{ij} + c_i\Big(\sum_t (\xi^{ij})_t\Big) + c_i^*\Big(\sum_t (\xi^{ij*})_t\Big),$$
$$((w^{ij})^{\mathrm{T}} \Phi(x_t)) + b^{ij} - y_i \leqslant \varepsilon - \xi_t^{ij},$$
$$y_i - ((w^{ij})^{\mathrm{T}} \Phi(x_t)) + b^{ij} \leqslant \varepsilon - \xi_t^{ij*}, \xi^{ij}, \xi^{ij*} \geqslant 0, i = 1, 2, \ldots \tag{17}$$

The relative results state that when the solution is not unique if and only if at least one of the following two conditions holds leading by the index sets $N_i, i = 1, 2, 3, 4$ from three equations in (17): $\sum_{i \in N_1} c_i = \sum_{i \in N_2} c_i^*$, $\sum_{i \in N_3} c_i^* = \sum_{i \in N_4} c_i$. They show that if the solution is not unique, primal (dual) solution does not necessarily imply uniqueness of the dual (primal) solution. They also show how to compute the threshold b^{ij} in Eq. (17) when the solution is unique, but when all support vectors are at bound the usual method for determining does not work. These statements give us a confidence on SVM in theory.

§ 4 Difficulties with nonlinear SVM

As the training data growing, the constraint part in P1 becomes larger and it is very memory-expensive, so several decomposition methods appear to decompose the constraint to fit in memory and solve the corresponding sub-quadratic programming iteration by iteration. The strategy to choose the better constrained subset to make training faster is a research topic[26-28].

The nonlinear kernel $k(\cdot,\cdot)\in\mathbf{R}^{m\times m}$ is fully dense (see § 2) causing the long CPU time to compute m^2 numbers. On the other hand,running out of memory after storing an $m\times m$ kernel matrix must be an exercise which is quite difficult for any computer. Computational complexity depends on m and over $O((m+1)^3)$ for almost any SVM's running. Furthermore,separating surface depends on almost the entire dataset and needs to store the entire dataset after solving the problem.

The choice of kernel function also influences the performance. In kernel methods discussed so far,the choice of kernel has a crucial effect on performance,i. e. ,if one does not choose the kernel properly,one will not achieve the excellent performance reported in many papers. Model selection techniques provide the principled ways to select a proper kernel. Usually,the candidates of optimal kernels are prepared using some heuristic rules, and the one which minimizes a given criterion is chosen. So,some determined factors arise and result in more errors.

For some classification problems, the numbers of data in different classes are unbalanced. Heace some researchers[7][10] have proposed to use different penalty parameters in SVM formulations,which isn't an ease to the correct ways.

§ 5 Open issues and challenges

Chances are that those readers who are still with us might be interested to hear how researchers have built on the above, applied the algorithm to real-world problems, and developed extensions. In this respect,several fields have emerged.

●Training methods for speeding up the quadratic program,such as the one described later in the installment of Trends & Controversies by Weston[9]. Speeding up the evaluation of decision function is of interest in a variety of applications, such as optical-character recognition (OCR). Unfortunately,as seen in § 4,the quadratic isn't often an easy exercise when a great deal of samples or high dimensional space's kernels are supplied.

●The choice of kernel functions, and hence of the feature space to work in,is of both theoretical and practical interest. It determines both the functional form of estimate and, via objective function of quadratic program, the type of regularization that is used to constrain the estimate. However, even though different kernels lead to different types of learning machines,the choice of kernel seems to be less crucial than it may appear at first sight. In OCR applications, for example, the kernels (Eqs. (6), (8)) lead to very similar performances and to strongly overlapping sets of support vectors.

●Although the use of SV methods in applications has only recently begun, the application developers have already reported state-of-the-art performances in a variety of applications in pattern recognition, regression estimation, and time series prediction. However, it is probably fair to say that we are still missing an application where SV methods significantly outperform any other available algorithm or solve a problem that has so far been impossible to tackle. For the latter, SV methods for solving inverse problems are a promising candidate.

● Using kernels for other algorithms emerges as an exciting opportunity for developing new learning techniques. The kernel method for computing dot products in feature spaces is not restricted to derive nonlinear generalizations of any algorithm that can be cast in terms of dot products. As a mere start, we decide to apply this most widely used algorithm for data analysis, principal component analysis (PCA). This leads to a kernel algorithm that performs nonlinear PCA by carrying out linear PCA in feature space. The method consists of solving a linear eigenvalue problem for a matrix whose elements are computed using the kernel function. The resulting feature extractors have the same architecture as SV machines. A number of researchers have since started to "kernel" various other linear algorithms.

●In nature, VC dimension is desirable because it is the foundation of SVM yet a few sets of popular function have been known well. Furthermore, the applicable researches of SVM are even later than the theatrical researches and all feasible methods to deal with these are promising works.

§ 6 Conclusion

What makes SVM attractive is: (a) the ability to condense the information contained in the training set, (b) the use of families of decision of relatively low VC-dimension, that is, minimization of the structural risk, (c) have unique solution, that is, have only one global minimization and would not trap into a local minimization, that is because the added term $\|w\|^2$ in object function in P1 translates the object function into a well form with only one local minimization[8], so the training speed in SVM is faster then traditional BP learning, (d) the added term $\|w\|^2$ in object function in P1 has more advantages such as making SVM equivalent to sparse approximation method[31] and equivalent to Regularization network[32-33], and (e) do not need to predefine the number of the support vectors, that is, the number of nodes in the hidden layer to the corresponding neural networks.

References

1 Boser, B. E., Guyon, I. M., Vapnik, V. N., A training algorithm for optimal margin classifiers, Proc.

Fifth Ann. Workshop Computational Learning Theory,New York:ACM Press,1992.

2 Vapnik,V. ,The Nature of Statistical Learning Theory,New York:Springer-Verlag,1995.

3 Schölkopf, B. , Smola, A. , Müller, K. R. , Nonlinear component analysis as a kernel eigenvalue problem,Neural Computation,1998,10:1299-1319.

4 Schölkopf,B. ,Burges,C. J. C. ,Smola,A. J. ,Advances in Kernel Methods-Support Vector Learning, United Kingdom:Cambrige University Press,1998.

5 Schölkopf,B. ,Support vector regression with automatic accuracy control,Proc. Eighth Int. Conf. Artificial Neural Networks,Perspectives in Neural Computing,Berlin:Springer-Verlag,1998.

6 Burges,C. J. C. ,Simplified Support Vector Decision Rules,Proc. 13th Int Conf. Machine Learning, Morgan Kaufmann,San Francisco,1996.

7 Smola,A. ,Schölkopf,B. ,From regularization operators to support vector kernels,In:M. Jordan,M. Kearns,and S. Solla,eds. ,Advances in Neural Information Processing Systems,MIT Press,1998.

8 Girosi,F. ,An equivalence between sparse approximation and support vector machines,AI memo No. 1606,MIT,Cambridge,Mass,1997.

9 Weston,J. ,Density Estimation Using Support Vector Machines,Tech. Report CSD-RT-97-23,Royal Holloway Univ. of London,1997.

10 Chang,C. C. ,Hsu,C. W. ,Lin,C. J. ,The analysis of decomposition methods for support vector machines,IEEE Trans. Neural networks,2000,11(4):1003-1008.

11 Chang, C. C. , Lin, C. J. , Training support vector classifiers: Theory and algorithms, Neural Computation,2001,13(9):2119-2147.

12 Chang, C. C. , Lin, C. J. , Training suport vector regression: Theory and algorithms. Neural Computation,2002,26:23-26.

13 Cortes,C. ,Vapnik,V. ,Support-vector network,Machine Learning,1995,20:273-297.

14 Crisp,D. J. ,Burges,C. J. ,A geometric interpretation of SVM classifiers,In:S. Solla,T. Leen,and Muller,K. R. Eds. ,Advances in Neural Information Processing Systems,2000,12:126-245.

15 Friedman,J. ,Another aproach to polychotomous classification,Technical Report,Department of Statistics,Stanford University Available,1996.

16 Hsu,C. W. ,Lin C. J. ,A comparison of methods for multi-class support vector machines,IEEE Transactions on Networks,2002,46:126-135.

17 Hsu,C. W. ,Lin C. J. ,A simple decomposition method for support vector machines,Machine Learning,2002,46:291-314.

18 Joachims,T. ,Making large-scale SVM learning practical,Machine Learning,1998,23:234-242.

19 Burges,T. ,Smola,A. J. (Eds.),Advances in Kernel Methods-Support Vector Learning,MA:MIT Press,1993.

20 Keerthi,S. S. ,Gilbert,E. G. ,Convergence of a generalized SMO algorithm for SVM classifier design, Machine Learning,2002,46:351-360.

21 Keerthi,S. S. ,Shevade,C. ,Bhattacharyya,A. S. ,et al. ,Improvements to Platt's SMO algorithm for SVM classifier design,Neural Computation,2001,13:637-649.

22 Knerr, S. , Personnaz, L. , Dreyfus, G. , Single-layer learning revisited: a stepwise procedure for building and training a neural network, In: J. Fogelman, ed. , Neurocompution: Algorithms, Architectures and Application,New York:Springer-Verlag,1996.

23 Krer,U. ,Pairwise classification and support vector machines,In:Schölkopf,B. C. ,Burges,J. C. and

Smola, A. J., eds., Advances in Kernel Methods-Support Vector Learning, MA: MIT Press, 1999.

24 Drucker, H., Wu, D., Vapnik, V. N., Support vector machines for span categorization, IEEE Trans. Neural Networks, 1999, 10: 1048-1054.

25 Mattera, D., Haykin, S., Support vector machines for dynamic reconstruction of a chaotic system, In: Advances in Kernel Methods-Support Vector Learning, MA: MIT Press, 1999.

26 Brown, M. P., Grundy, W. N., Lin, C. J., Knowledge-based analysis of microarray gene expression data using support vector machines, Proc. National Academy Science, 2000, 97, (1): 262-267.

27 Furey, T., Cristianini, N., Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics, 2000, 16: 906-914.

28 Zien, A., Engineering support vector machine kernels that recognize translation initiation sites in DNA, Bioinformatics, 2000, 16: 799-807.

29 Hsu, C. W., Lin, C. J., A simple decomposition method for support vector machines, Machine Learning, 1999, 46: 291-314.

30 Haussler, D., Convolution kernels on discrete structures, UC Santata Cruzzy, Technical Report, No. UCSC-CRL-99-10, 1999.

31 Watkins, C., Dynamic alignment kernels, In: A. J. Smola, P. L. Bartlett, B. Schölkopf, et, al., eds., Advances in Large Margin Classifiers, MA: MIT Press, 2000.

32 Burges, C. J., A tutorial way on support vector machines for pattern recognition, Knowledge Discovery and Data Mining, 1998, 2(2): 121-167.

33 Smola, A., Schölkopf, B., A tutorial on support vector regression, Statistics and Computing, 2001, 12: 212-226.

Institute of Industrial Process Control, Zhejiang Univ., Hangzhou, 310027, China. Email: shyue@iipc.zju.edu.cn

* Dept. of Computer Sci. and Inform. Eng., Cheng Kung Univ., Taiwan, China.