How Many Neighbors To Consider in Pattern Pre-selection for Support Vector Classifiers?

Hyunjung Shin and Sungzoon Cho
Department of Industrial Engineering, Seoul National University
San 56-1, Shillim-Dong, Kwanak-Gu, 151-744, Seoul, Korea
Email: {hjshin72, zoon}@snu.ac.kr

Abstract— **Training support vector classifiers (SVC) requires large memory and long cpu time when the pattern set is large. To alleviate the computational burden in SVC training, we previously proposed a preprocessing algorithm which selects only the patterns in the overlap region around the decision boundary, based on neighborhood properties [8], [9], [10]. The k-nearest neighbors' class label entropy for each pattern was used to estimate the pattern's proximity to the decision boundary. The value of parameter k is critical, yet has been determined by a rather ad-hoc fashion. We propose in this paper a systematic procedure to determine k and show its effectiveness through experiments.**

I. Introduction

In SVC quadratic programming (QP) formulation, the dimension of kernel matrix ($M \times M$) is equal to the number of training patterns(M). Most standard QP solvers have time complexity O(M^3): MINOS, CPLEX, LOQO and MATLAB QP routines. In order to solve a large scale SVC QP problem, decomposition methods or iterative methods have been suggested which break down the large QP problem into a series of smaller QP problems: Chunking, SMO, SVM$^{\text{light}}$ and SOR [4], [6]. The general time complexity of those methods is approximately (*the number of iterations*)$\cdot$ O($Mq+q^3$) where q is the size of the working set. Of course, "the number of iterations" increases as M increases.
One way to circumvent this computational burden is to select only the training patterns, in advance, that are more likely to be support vectors. The reduced timing data set leads to reduction in training time (see Fig. 1). In a classification problem, the support vectors tend to be distributed near the decision boundary. A considerable amount of research efforts have been made to select the patterns near the decision boundary [1], [2], [5], [7].

The approach we recently proposed selected the patterns near the decision boundary based on the neighborhood properties [8]. First, a pattern located near the decision boundary tends to have more heterogeneous neighbors. The degree of class heterogeneity can be quantified using the entropy value of neighbors' class labels. The degree of proximity to the decision boundary thus can now be estimated by neighbors' label entropy. Patterns with a large neighbors' entropy value are considered to be close to the decision boundary thus selected for training. Among them, however, "overlap" patterns are also present which result from two class-distributions' overlapping. Of course, genuine noisy patterns are also included. These patterns have to be identified and removed. The second property dictates that an overlap or a noisy pattern tends to belong to a different class from its neighbors. Potential overlap patterns can be eliminated by the ratio of the neighbors whose label matches that of the pattern. A smaller ratio indicates that the pattern is potentially incorrectly labeled. This two step procedure reduced the number of patterns significantly, thus reduced the training time while keeping the SVC accuracy intact. Table I compares the numbers of patterns and computing times of SVC with all patterns vs SVC with the selected patterns for two synthetic data sets [9]. Overall computing time (selection plus training) was significantly reduced, 30 times for sine function and 113 times for XOR, while the generalization error did not increase. The algorithm worked well since "important" patterns located near the decision boundary were selected for training as shown in Fig. 2 for sine function problem.

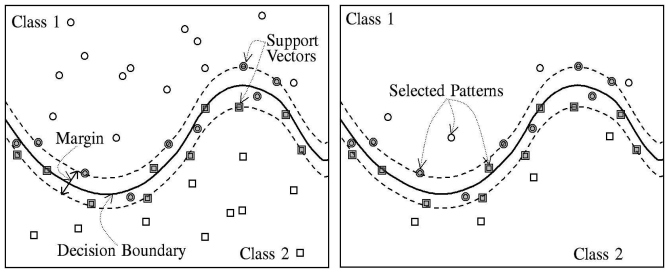


Fig. 1. Pattern selection: a large training set shown in (a) is condensed to a small training set (b) which is composed of only potential support vectors.

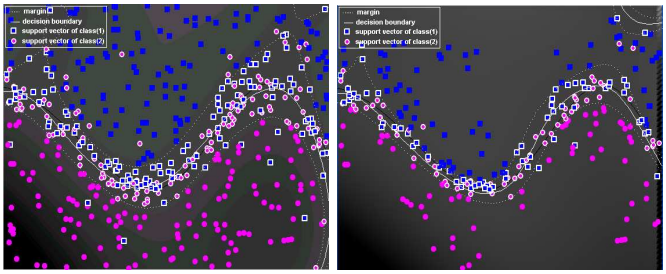
One may compute the neighbors' label entropy for the patterns near the decision boundary only, not all training patterns since the neighbors of the pattern located near the decision boundary tend to be located near the decision boundary as well. This lazy evaluation turns the time complexity from O(M^2) to O(vM), where v is the number of patterns in the "overlap" region around the decision boundary that is enclosed by the overlap patterns located farthest from the boundary. In most practical problems, $v < M$ holds. A pattern is assumed to belong to the region if its k nearest neighbors belong to more than one class, or its k nearest neighbors' label entropy is positive. Note that parameter k

0-7803-7898-9/03/$17.00 ©2003 IEEE

TABLE I

SVC TRAINING RESULT WITH ALL / SELECTED PATTERNS (SEE [9])

	Sine Function (poly, degree=4, C=100)		**Continuous XOR** (rbf, width=1, C=100)	
	All	Selected	All	Selected
Num. of Trn. Patterns	500	264	600	180
Num. of SVs	250	136	167	84
Test Error (%)	13.33	13.33	9.67	9.67
SVC Trn. Time (sec.)	267.76	8.79	454.83	3.85
Pattern Sel. Time (sec.)	-	0.17	-	0.21



(a) ***SVC with all patterns*** **(b)** ***SVC with selected patterns***

Fig. 2. Patterns and SVC decision boundaries of sine Function problem: decision boundary is depicted as a solid line and the margins are defined by the dotted lines in both sides of it. Support vectors are outlined.

determines the extent with which neighbors are defined. In our previous studies, it was determined in a rather ad-hoc way.

We propose in this paper a systematic procedure to determine k. First, the number of patterns located in the "overlap" region, v, is estimated. Second, we find k such that the number of the patterns with a positive k nearest neighbors' label entropy is larger than v. We also show through two experiments that the proposed method estimates v with a high accuracy and that the number of the patterns selected using the method is large enough to result in a comparable classification accuracy.

In section 2, we present the procedure to identify the patterns that are likely to lie in overlap region. In section 3 and section 4, we provide empirical results supporting our approach. In the last section, we conclude the paper with the discussion of the limitations and future work.

II. ESTIMATING OVERLAP PATTERN SET $\mathbf{V}$ WITH $\mathbf{B}_k$

In this section, we propose a procedure to identify a subset of the training pattern set $\mathbf{D}$ that matches the overlap region $\mathbf{R}$ as closely as possible. First, we give definitions of classifier $f(\vec{x})$, training pattern set $\mathbf{D}$, overlap pattern set $\mathbf{V}$ and overlap region $\mathbf{R}$ as well as positive k nearest neighbors' entropy pattern set $\mathbf{B}_k$. Second, some properties of $\mathbf{B}_k$ as well as the proposed procedure are presented. Finally, an estimate of the cardinality of $\mathbf{V}$ is presented.

Consider a two-class classification problem whose classes

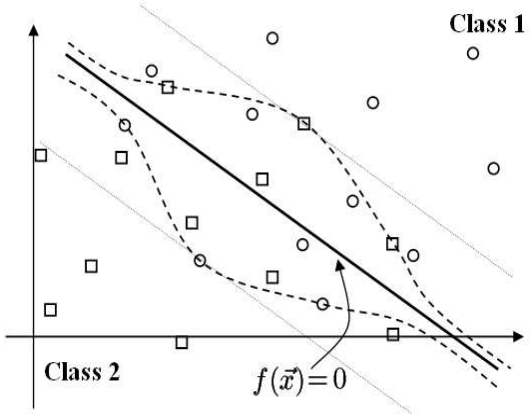


Fig. 3. Two class classification problem where circles belong to class 1 while squares belong to class 2. The area enclosed by the two dotted lines comprise the overlap area.

are C_1 and C_2 (see Fig. 3), with classifier $f(\vec{x})$ such that

$$f(\vec{x}) = \begin{cases} \vec{x} \rightarrow C_1 & \text{if} \quad f(\vec{x}) > 0, \\ \vec{x} \rightarrow C_2 & \text{if} \quad f(\vec{x}) < 0, \end{cases} \tag{1}$$

and $f(\vec{x}) = 0$ is its decision boundary. Let $\mathbf{D}$ denote the set of training patterns. Let us define "overlap patterns" as the patterns that are located in the "other" side of the decision boundary since the class distributions overlap. For simplicity, we will consider genuine noisy patterns as overlap patterns. They are shown in Fig. 3 as squares located above $f(\vec{x}) = 0$ and circles located below $f(\vec{x}) = 0$. Let $\mathbf{R}$ denote a hypothetical region where the overlap patterns reside, the area enclosed by the dotted lines in Fig. 3. Note that $\mathbf{R}$ contains not only the overlap patterns, but also the "close non-overlap" patterns, those patterns that are located close to the decision boundary, yet in the "right" side of the decision boundary. Let $\mathbf{V}$ denote the intersection of $\mathbf{D}$ and $\mathbf{R}$, i. e. the subset of $\mathbf{D}$ which comprises overlap patterns and close non-overlap patterns. There are six patterns in class 1 side and another six patterns in class 2 side in Fig. 3. The cardinality of $\mathbf{V}$ is denoted as v.

Now, let $\mathbf{B}_k$ denote a subset of $\mathbf{D}$ whose elements have positive k nearest neighbors' entropy values (see Fig. 4):

$$\mathbf{B}_k = \{\vec{x} \mid \text{Neighbors\_Entropy}(\vec{x}, k) > 0,\ \vec{x} \in \mathbf{D}\}. \tag{2}$$

Let us consider how k affects $\mathbf{B}_k$ and pattern selection based on it. Too large a value of k results in *excessive inclusion* of the training patterns. In other words, too many patterns are selected. If $k = M - 1$, then $\mathbf{B}_k$ becomes $\mathbf{D}$. Suppose that pattern $\vec{x}$ belongs to C_1. Then its LabelProbability$(\vec{x}, k)$ is

$$P_1 = \frac{m_1 - 1}{m_1 + m_2 - 1},$$
$$P_2 = \frac{m_2}{m_1 + m_2 - 1},$$

where m_j denotes the number of patterns belonging to C_j, $(j = 1, 2)$. Thus, we have $P_j < 1$ for all j's. If pattern $\vec{x}$

LabelProbability($\vec{x}, k$) {

/* For $\vec{x}$, calculate the label probabilities of $k\mathbf{NN}(\vec{x})$ over J classes, $\{C_1, C_2, \ldots, C_J\}$, where $k\mathbf{NN}(\vec{x})$ is defined as the set of k nearest neighbors of $\vec{x}$. */

$k_j = |\{\vec{x}' \in C_j | \vec{x}' \in k\mathbf{NN}(\vec{x})\}|, \ \ j = 1, \ldots, J.$

return $\left(P_j = \frac{k_j}{k}, \ \forall j\right)$.

}

Neighbors_Entropy($\vec{x}, k$) {

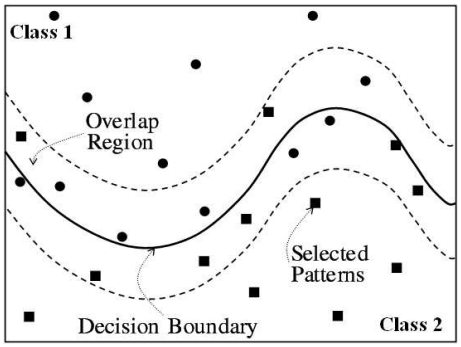
/* Calculate the neighbors-entropy of $\vec{x}$ with its nearest neighbors' labels. In all calculations, $0\, log_J \frac{1}{0}$ is defined to be 0. */

Do **LabelProbability($\vec{x}, k$)**.

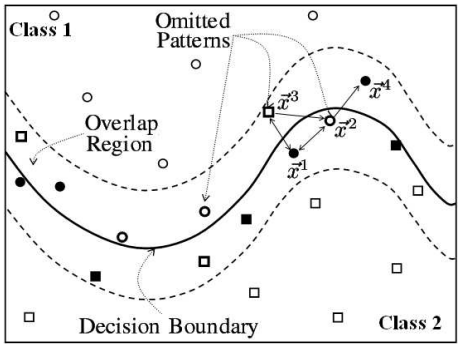
return $\left(\sum_{j=1}^{J} P_j \cdot log_J \frac{1}{P_j}\right)$.

}

Fig. 4. LabelProbability and Neighbors_Entropy



(a) *Excessive inclusion: k=M-1*



(b) *Insufficient inclusion: k=2*

Fig. 5. Effect of k on $\mathbf{B}$: solid dots and squares belong to $\mathbf{B}$.

belongs to C_2, both P_1 and P_2 are less than 1. Remember that Neighbors_Entropy($\vec{x}, k$) in Fig. 4 is always positive unless j exists such that $P_j = 1$. Therefore, all the patterns in training set $\mathbf{D}$ have positive Neighbors_Entropy values, regardless of their location in the input space, thus become a member of $\mathbf{B}_{M-1}$. Every pattern from $\mathbf{D}$ is selected for training (see Fig. 5(a)). Too small a value of k, e. g. $k = 2$, on the other hand, results in *insufficient inclusion* of the patterns within the overlap region. Consider patterns $\vec{x}^1$, $\vec{x}^2$ and $\vec{x}^3$ in Fig. 5(b), lying within the overlap region. They all belong to overlap data set $\mathbf{V}$, but $\vec{x}^2$ and $\vec{x}^3$ do not belong to $\mathbf{B}_2$ while $\vec{x}^1$ does. First, $\vec{x}^1$ belongs to $\mathbf{B}$ since its two nearest neighbors $\vec{x}^2$ and $\vec{x}^3$ belong to different classes, which results in $P_1 = P_2 = 1/2$ and Neighbors_Entropy($\vec{x}^1$, k) becomes 1. Second, the two nearest neighbors of $\vec{x}^2$, $\vec{x}^1$ and $\vec{x}^4$, both belong to class C_1, which results in $P_1 = 1$, $P_2 = 0$ and Neighbors_Entropy($\vec{x}^2$, k) is 0. So, $\vec{x}^2$ does not belong to $\mathbf{B}$. Third, for the same reason, either does not $\vec{x}^3$. The patterns in the overlap region is critical to SVC training, since they are likely to be support vectors. Therefore, the exclusion of them could degrade the SVC prediction accuracy.

In short, $\mathbf{B}_k$ larger than $\mathbf{V}$ merely increases the SVC training time by introducing redundant training patterns. On the contrary, $\mathbf{B}_k$ smaller than $\mathbf{V}$ could degrade the SVC accuracy. Therefore, our objective is to find the smallest $\mathbf{B}_k$ that covers $\mathbf{V}$. The following property of $\mathbf{B}_k$ results in a simple procedure.

Lemma 1 $\mathbf{B}_k \subseteq \mathbf{B}_{k+1}$ for $k = 2, \cdots, M-2$.

Proof: Denote P_j^k as the probability that k_j out of k nearest neighbors belong to class C_j. If $\vec{x} \in \mathbf{B}_k$, then it means Neighbors_Entropy($\vec{x}, k$) > 0. A positive Neighbors_Entropy is always accompanied with $P_j^k = \frac{k_j}{k} < 1, \ \forall j$. Therefore,

$$k_j < k, \ \forall j. \tag{3}$$

Adding 1 to both sides yields

$$(k_j + 1) < (k + 1), \ \forall j. \tag{4}$$

Suppose $(k+1)^{th}$ nearest neighbor belongs to C_{j^*}. Then, for j^*, $k_{j^*} + 1 < k + 1$ holds while for $j \neq j^*$, $k_j < k + 1$ holds. Therefore, both $P_{j^*}^{k+1} < 1$ and $P_j^{k+1} < 1, \ \forall j \neq j^*$. We have Neighbors_Entropy($\vec{x}, k+1$) > 0 which indicates $\vec{x} \in \mathbf{B}_{k+1}$. ■

From Lemma 1, it follows that b_k, the cardinality of $\mathbf{B}_k$, is an increasing function of k. Thus optimal k, k^*, is computed as

$$k^* = \min\{k \mid b_k \geq v, \ k = 2, \ \ldots, \ M-1 \ \}. \tag{5}$$

Now, we need to estimate v. Every training pattern that we encounter can be regarded as an independently sampled pattern from a training data distribution. Then, the probability that v patterns of M training patterns fall within region $\mathbf{R}$ is given by the binomial law,

$$Pr(v) = \frac{M!}{v!(M-v)!} \left(P_{\mathbf{R}}(\vec{x})\right)^v \left(1 - P_{\mathbf{R}}(\vec{x})\right)^{M-v}. \tag{6}$$

where $P_{\mathbf{R}}(\vec{x})$ denotes the probability that a pattern $\vec{x}$ lies in $\mathbf{R}$. We now can calculate v as

$$v = M P_{\mathbf{R}}(\vec{x}). \tag{7}$$

Here $P_{\mathbf{R}}(\vec{x})$ of $\vec{x}$ can be described as

$$P_{\mathbf{R}}(\vec{x}) = \sum_{j=1}^{2} P(\vec{x} \in \mathbf{R}, C_j), \tag{8}$$

where $P(\vec{x} \in \mathbf{R}, C_j)$ is the joint probability of $\vec{x}$ belonging to class C_j and lying in $\mathbf{R}$. We divide the region $\mathbf{R}$ into $\mathbf{R}_1$ and $\mathbf{R}_2$ as follows:

$$\begin{array}{rcl} \mathbf{R}_1 & = & \{\vec{x} \in \mathbf{R} \mid f(\vec{x}) \geq 0\}, \\ \mathbf{R}_2 & = & \{\vec{x} \in \mathbf{R} \mid f(\vec{x}) < 0\}. \end{array} \tag{9}$$

Eq. (8) can be rewritten as

$$\begin{aligned} P_{\mathbf{R}}(\vec{x}) &= P(\vec{x} \in \mathbf{R}, C_1) + P(\vec{x} \in \mathbf{R}, C_2) \\ &= P(\vec{x} \in \mathbf{R}_1 \cup \mathbf{R}_2, C_1) + P(\vec{x} \in \mathbf{R}_1 \cup \mathbf{R}_2, C_2) \\ &= \Big(P(\vec{x} \in \mathbf{R}_1, C_2) + P(\vec{x} \in \mathbf{R}_2, C_1) \Big) \\ &+ \Big(P(\vec{x} \in \mathbf{R}_1, C_1) + P(\vec{x} \in \mathbf{R}_2, C_2) \Big). \end{aligned} \tag{10}$$

The first parenthesis and second parenthesis denote the probabilities that patterns located in $\mathbf{R}$ are incorrectly and correctly classified, respectively. If $\mathbf{R}_1$ and $\mathbf{R}_2$ contain roughly the same number of correct and incorrect patterns, the probabilities of the two parentheses become same. Since all the overlap patterns were included in $\mathbf{R}$, the first parenthesis actually refers to the misclassified error of classifier $f(\vec{x})$, or $P(\text{error})$. Now, Eq. (10) can be simplified as

$$P_{\mathbf{R}}(\vec{x}) = 2P(\text{error}), \tag{11}$$

and Eq. (7) becomes

$$v = 2MP(\text{error}). \tag{12}$$

Now, the procedure to determine the optimal k value is as follows:

1) Apply *1*-NN rule over training set $\mathbf{D}$.
2) Estimate $P(\text{error})$ with $\hat{P}(\text{error})$, the training error rate of 1).
3) Calculate $\hat{v}$ according to Eq. (12): $\hat{v} = 2M\hat{P}(\text{error})$.
4) Find k^* according to Eq. (5): $k^* = \min\{k \mid b_k \geq \hat{v},\ k = 2,\ \ldots,\ M-1\ \}$.

Reasons for using *1*-NN rule to estimate $P(\text{error})$ include its simplicity and computational efficiency.

III. Synthetic Data Experiments

In the first experiment, we examined whether the proposed method gave a reasonably accurate estimation for v. A total of 1,000 (=M) patterns, 500 from each class, were randomly generated from a pair of two-dimensional uniform distributions:

$$\begin{aligned} C_1 = &\ \left\{ \vec{x} \mid U\left(\begin{bmatrix} -1 \\ (\ 0 - \frac{1}{2}\frac{v}{1000}) \end{bmatrix} < \vec{x} < \begin{bmatrix} 1 \\ (1 - \frac{1}{2}\frac{v}{1000}) \end{bmatrix} \right) \right\}, \\ C_2 = &\ \left\{ \vec{x} \mid U\left(\begin{bmatrix} -1 \\ (-1 + \frac{1}{2}\frac{v}{1000}) \end{bmatrix} < \vec{x} < \begin{bmatrix} 1 \\ (0 + \frac{1}{2}\frac{v}{1000}) \end{bmatrix} \right) \right\}. \end{aligned}$$

TABLE II
Estimation of v for various overlap degrees

v	100	200	300	400	500	600	700	800	900	1000
$\hat{v}$	112	202	334	414	512	602	706	806	882	942
b_{k^*}	115	209	339	429	518	606	708	816	906	972
k^*	5	4	5	4	5	5	5	5	5	5

We used 10 training pattern sets corresponding to 10 different numbers of overlap patterns, i.e. v= 100, 200, $\cdots$, 900, 1000.

Table II provides the estimation results for various values of v. The second row shows the estimated values of $\hat{v}$. They are almost identical to the true values of v. The proposed method gave reasonably accurate estimation of v. The last two rows show the smallest b_k larger than $\hat{v}$, and the corresponding value of k. Approximately, k=5 seems to cover the overlap region regardless of the different degrees of overlap. The optimal value of k is likely to be dependent on the underlying distribution rather than the degree of overlap itself.

The second one is a continuous XOR problem. The patterns of two classes were defined as follows:

$$\begin{aligned} C_1 = &\ \left\{ \vec{x} \mid \vec{x} \in C_{1A} \cup C_{1B},\ \begin{bmatrix} -3 \\ -3 \end{bmatrix} \leq \vec{x} \leq \begin{bmatrix} 3 \\ 3 \end{bmatrix} \right\}, \\ C_2 = &\ \left\{ \vec{x} \mid \vec{x} \in C_{2A} \cup C_{2B},\ \begin{bmatrix} -3 \\ -3 \end{bmatrix} \leq \vec{x} \leq \begin{bmatrix} 3 \\ 3 \end{bmatrix} \right\} \end{aligned}$$

where C_{1A}, C_{1B}, C_{2A} and C_{2B} were

$$\begin{aligned} C_{1A} = &\ \left\{ \vec{x} \mid N\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right) \right\}, \\ C_{1B} = &\ \left\{ \vec{x} \mid N\left(\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right) \right\}, \\ C_{2A} = &\ \left\{ \vec{x} \mid N\left(\begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right) \right\}, \\ C_{2B} = &\ \left\{ \vec{x} \mid N\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right) \right\}. \end{aligned}$$

A total of 600 training patterns, 300 from each class, were generated: There are about 33% training patterns in the overlap region (v=199). The density of patterns gets sparser when it goes closer to the decision boundary. A total of 1000 test patterns were generated from the statistically same distributions as in its training sets.

The proposed method estimated v as 208 ($\hat{v}$=208). And the value of k was set as 5 since b_5 was the minimum over 208 (k^*=5 and b_{k^*}=217). See Fig. 6. In order to test whether the selected pattern set taken from B_{k^*} will give rise to a reasonable SVC performance, we generated 29 selected pattern sets corresponding to k=2, ..., 30, and then we computed the SVC test error rates of the 29 sets. The 14.1% reference test error rate was obtained from the SVC trained with all 600 training patterns, among which 162 were picked as support vectors. We set the SVC error tolerance value as C=20 and used the RBF kernel with width parameter σ=0.5. The parameter values were fixed over all 30 SVCs. Fig. 6

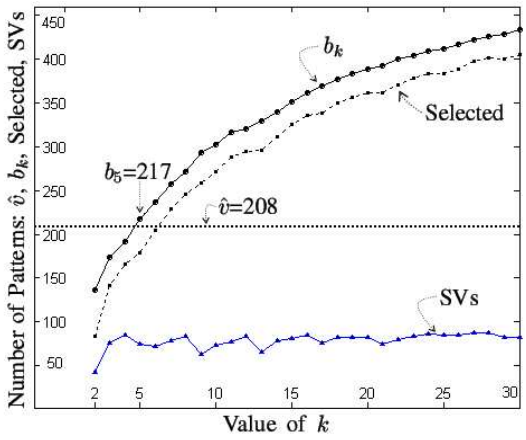


Fig. 6. Number of Patterns: $\hat{v}$, b_k, and SVs

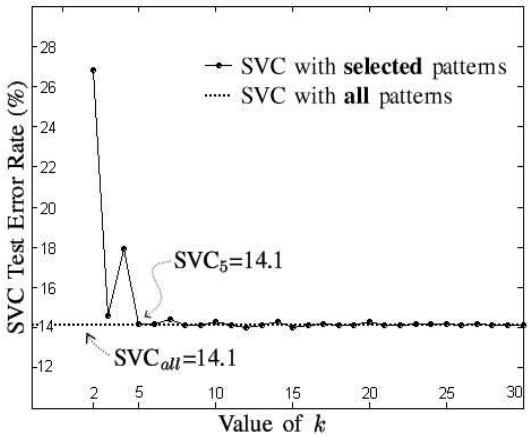
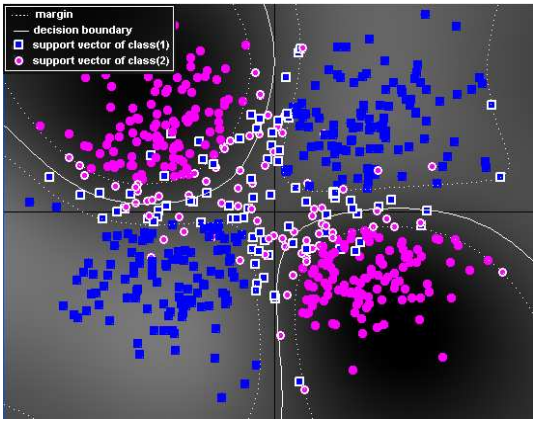
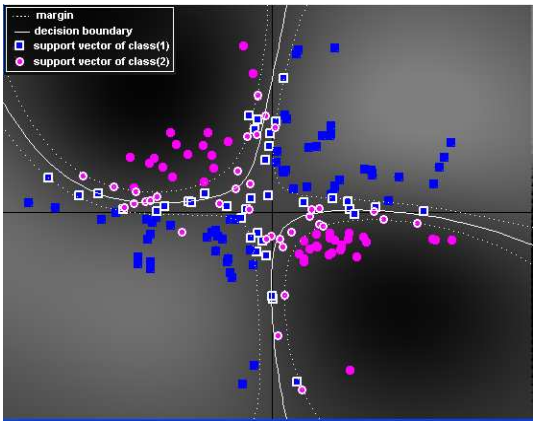


Fig. 7. SVC test error rate



(a) ***All patterns***



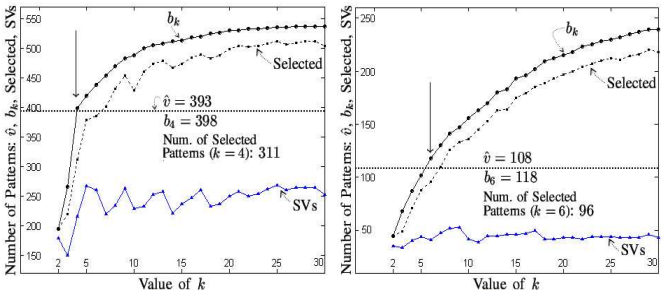
(b) ***Selected patterns (k=5)***

Fig. 8. Patterns and SVC decision boundaries: decision boundary is depicted as a solid line and the margins are defined by the dotted lines in both sides of it. Support vectors are outlined.

shows that, when v is fixed, the increase of k induced the increase of b_k. The number of selected patterns was slightly less than b_k, but it also gradually increased almost parallel to the curve of b_k. The number of support vectors was also given at the bottom. For $k > 5$, it converged to about 78, which is only half of the 162 SVs that were picked when trained with the full training data set. That is, only a subset of training patterns affected SVC training regardless of the number of training patterns. Meanwhile, the reason why only 78 SVs were adopted can be explained by "Neighbors_Match criterion" that identifies and removes those patterns that are suspected to be overlap patterns [9]. Those overlap patterns were adopted as SVs in the original SVC by its error tolerance parameter, but they hardly contributed to margin constitution. Fig. 7 displays the test error rates for 30 different SVCs. The SVC performance was stabilized for k larger than 5 at which the test error rate was 14.1%. It is almost same as the reference test error rate. A larger pattern set than $\mathbf{B}_5$ did not lead to better SVC performance since they included the redundant patterns which did not contribute to SVC training. An evidence can be found from the number of support vectors in Fig. 6. From k=5 to k=30, the number of support vectors did not increase much from 75 to 83 while the number of the selected patterns increased by more than two times from 179 to 405. Finally, Fig. 8 shows the decision boundaries and margins of the SVCs (a) with all patterns and (b) with the selected patterns with k=5. Note that the two decision boundaries are quite similar. The selected patterns from $\mathbf{B}_5$ were sufficient enough to result in the same classification accuracy as the original SVC.

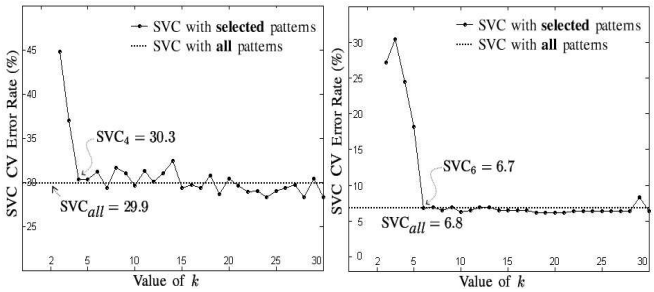
IV. Real World Data Experiments

We also applied the proposed approach to two real world datasets [11]: Pima Indian Diabetes and Wisconsin Breast Cancer. We conducted 5-fold cross validation (CV). According to Eq. (12), v values for the two datasets were estimated as 393 and 108 from P(error)=32.0% and P(error)=9.9%, respectively. The value of k was determined by b_k which was just larger than $\hat{v}$: $k = 4$ in Pima Indian Diabetes and $k = 6$ in Wisconsin Breast Cancer (see Fig. 9). We chose a quadratic polynomial SVC kernel and the error tolerance parameter C=100 for Pima Indian Diabetes, and a cubic polynomial kernel and C=5 for Wisconsin Breast Cancer. Fig. 10 depicts the average SVC CV error rates. In Pima Indian Diabetes, the SVC performance was stabilized at about 30.0% for k larger than 4, and in Wisconsin Breast Cancer, 6.7% for k larger



(a) ***Pima Indians Diabetes*** **(b)** ***Wisconsin Breast Cancer***

Fig. 9. Number of patterns: $\hat{v}$, b_k, and SVs



(a) ***Pima Indians Diabetes (k=4)*** **(b)** ***Wisconsin Breast Cancer (k=6)***

Fig. 10. SVC CV error rates

than 6. The results are similar to those from the synthetic data experiments. Table III compares the average execution times and the performances of SVC with all patterns vs SVC with the selected patterns. In both datasets, the average SVC training time was reduced from 203.91 (sec) to 27.86 (sec), and from 2.14 (sec) to 0.03 (sec), respectively, but on the other hand, the SVC performances were almost reserved.

V. Conclusion

In this paper, we proposed how to determine parameter k, the number of neighbors, to complete the pattern selection algorithm. We presented a definition of the overlap region $\mathbf{R}$ first, followed by a derivation of v, the number of patterns in the overlap region. Finally, by *1*-NN rule estimation for v in $\mathbf{R}$, we determined k. Through the experiments on two synthetic and two real world problems, we justified the proposed method.

Currently, we just applied the proposed method to a two-class problem under the assumption that $\mathbf{R}_1$ and $\mathbf{R}_2$ contain roughly a same number of correct and incorrect patterns. Therefore, further extension for more general cases will be conducted.

Another candidate for future work is concerned with the input dimensionality. In the proposed algorithm, a pattern's proximity to the decision boundary is estimated by the diversity of its k neighbors' class labels. A higher dimensionality usually requires a larger k value, or more neighbors to be considered. The error rate of 1-NN rule in a higher dimensional space increases, but in a much less degree [3]. Combination of these gives rise to an underestimation of the number of the patterns in the overlap region by the 1-NN rule (see Eq. (12)) which in turn results in an underestimation of k. A preliminary experiment involving MNIST datasets with input vectors of 784 dimension confirms the observation. The proposed method resulted in a k value less than 10 while a much better SVC performance was obtained with a k value larger than 50 [9]. An extension or a modification to the proposed approach is deemed necessary to handle the high dimensionality.

TABLE III
SVC Training Results with All / Selected Patterns

	Num. of Trn. patterns	Num. of SVs	Pattern Sel. time (sec)	SVC trn. time (sec)	SVC CV error (%)
Pima Indian Diabetes					
All	615	330	-	203.91	29.9
Selected ($k = 4$)	311	216	0.24	27.86	30.3
Wisconsin Breast Cancer					
All	546	87	-	2.14	6.8
Selected ($k = 6$)	96	41	0.10	0.03	6.7

References

[1] Almeida, M. B., Braga, A. and Braga J. P., "SVM-KM: speeding SVMs learning with a priori cluster selection and k-means," *Proc. of the 6th Brazilian Symposium on Neural Networks*, pp. 162–167, 2000.

[2] Choi, S. H. and Rockett, P., "The Training of Neural Classifiers with Condensed Dataset," *IEEE Transactions on Systems, Man, and Cybernetics-PART B: Cybernetics*, vol. 32, no. 2, pp. 202–207, 2002.

[3] Ferri, F. J., Albert, J. V. and Vidal. E., "Considerations About Sample-Size Sensitivity of a Family of Edited Nearest-Neighbor Rules," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 29, no. 4, pp. 667–672, 1999.

[4] Hearst, M. A., Scholkopf, B., Dumais, S., Osuna, E., and Platt, J., 1998. "Trends and Controversies - Support Vector Machines," *IEEE Intelligent Systems*, vol. 13, pp. 18–28, 1997.

[5] Lyhyaoui, A., Martinez, M., Mora, I., Vazquez, M., Sancho, J. and Figueiras-Vaidal, A. R., "Sample Selection Via Clustering to Construct Support Vector-Like Classifiers," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1474–1481, 1999.

[6] Platt, J. C. "Fast Training of Support Vector Machines Using Sequential Minimal Optimization," *Advances in Kernel Methods: Support Vector Machines*, MIT press, Cambridge, MA, pp. 185–208, 1999.

[7] Shin, H. J. and Cho, S., "Pattern Selection Using the Bias and Variance of Ensemble," *Journal of the Korean Institute of Industrial Engineers*, vol. 28, no. 1, pp. 112–127, 2002.

[8] Shin, H. J. and Cho, S., "Pattern Selection For Support Vector Classifiers," *Proc. of the 3rd International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, Manchester, UK, pp. 469–474, 2002.

[9] Shin, H. J. and Cho, S., "Fast Pattern Selection for Support Vector Classifiers," *Proc. of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, Seoul, Korea, in press, 2003.

[10] Shin, H. J. and Cho, S., "Fast Pattern Selection Algorithm for Support Vector Classifiers: Time Complexity Analysis," *Proc. of the 4th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, Hong Kong, China, in press, 2003.

[11] http://www.ics.uci.edu/∼mlearn/