Support Vector Based Prototype Selection Method for Nearest Neighbor Rules

Yuangui Li, Zhonghui Hu, Yunze Cai, and Weidong Zhang

Department of Automation, Shanghai Jiaotong University
1954, Huashan Road, Xuhui, Shanghai, 200030, P. R. China
{li_yuangui, huhzh, yzcai, wdzhang}@sjtu.edu.cn.edu

Abstract. The Support vector machines derive the class decision hyper planes from a few, selected prototypes, the support vectors (SVs) according to the principle of structure risk minimization, so they have good generalization ability. We proposed a new prototype selection method based on support vectors for nearest neighbor rules. It selects prototypes only from support vectors. During classification, for unknown example, it can be classified into the same class as the nearest neighbor in feature space among all the prototypes. Computational results show that our method can obtain higher reduction rate and accuracy than popular condensing or editing instance reduction method.

1 Introduction

For classification problems, complete statistical knowledge regarding the conditional density functions of each class is rarely available, which precludes the application of the optimal Bayes classification methods, while the nearest neighbor(NN) rule and its extension to k neighbors (or k-NN rule) have been in practice one of the most widely used non-parametric classifiers. The advantage of NN rule lies in that it combines its conceptual simplicity with the fact that its asymptotic error rate is conveniently bounded in terms of the optimal Bayes error [1]. However, the main problems of the NN rules lie that it is computationally expensive and the storage requirement is large for large problems because it stores all the training examples in memory and distances between new instance and all the training points is required to be computed to find the nearest neighbor in classifying process; and it is intolerant to noisy instance and irrelevant attributes. Many researches on prototype selection have been done in order to reduce the training set, reduce the effect of noise on accuracy, and obtain the same classification ability as using the whole training set [2-4].

Two different families of prototype selection methods exist in the literature. First, the condensing or reducing algorithm aims at selecting the minimal subset of prototypes that lead to the same performance as using the whole training set. Second, editing algorithm eliminates noisy examples from the original set and "cleans" possible overlapping among classes. The recent condensing algorithm is Minimal Consistent Set(MCS) method proposed by Dasarathy[5] and Dasarathy conjectured MCS was the minimal training-set consistent subset, but the counter-examples to this claim have been found by Kuncheva and Bezdek[6]. The difficulty of condensing algorithm is

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3610, pp. 528 – 535, 2005.

that the noisy examples are preferred to be selected into prototype set, which harms the accuracy of result classifier. For editing algorithms, it is observed that the asymptotically optimal edited NN-rule, such as well known Multi-edit algorithm, can lead to arbitrarily bad classification result if the number of prototypes is not large enough compared to the intrinsic dimension of feature space[7]. Furthermore the editing algorithm can't reduce the training set effectively. Dasarathy[7] found that the synergy exploitation of condensing and editing algorithm could make the best result on balance of instance reduction with classification accuracy. So an effective prototype selection algorithm should be able to both remove the noise and overlapping out of prototype set and obtain an as small as possible prototype set.

The support vector machine (SVM) is a new kind of learning machine proposed by Vapnik in 1995[8]. It is derived from statistical learning theory and VC-dimension theory [9-12], and has become another research hotspot following neural network. The remarkable advantage of SVM is that it is induced according to the principle of structural risk minimization, so it performs good generalization ability, especially for small sample problems. The decision surface of SVM is parameterized by a set of support vectors and a set of corresponding weights, which indicates that support vectors have the key patterns to define the decision boundaries. So it is possible to develop new prototype selection base on support vectors. Vishwanathan and Murty[15] proposed data reduction method using multi-category proximal SVM, but it simply selected the support vectors with Langrage multipliers larger than 0 and less than the bound. They only indicated that it is feasible to select prototypes for NN with SVM and didn't compare the performance with common instance reduction method.

In order to select prototypes based on support vectors, we should obtain SVM first, why not use SVM to classify new examples? LeeCun et al. [16] found that the classification speed of SVM is substantially slower than that of neural networks, especially for large problems. That is because too many support vectors is required to express the decision boundary and increase the complexity of decision function. To address this problem, Burges[13-14] proposed simplified SVM, which used a new reduced vector set to approximate the decision rule decided by all the support vectors so as to reduce the complexity of SVM and assure the loss in generalization performance is acceptable, and in some cases, the reduced vector set can be computed analytically. But Burges' method is too complex.

This paper is organized as follows. In section 2, the different importance in deciding classification hyper planes between 3 types of support vectors was analyzed. In section 3 we introduced our prototype selection method based on support vectors. Computational results are presented in section 4 to compare performance of our method with that of common instance reduction methods. Section 5 concludes our work.

2 Support Vectors and Decision Hyper Planes of SVM

Suppose that there exists a given training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l} \in X^n \times R$, where X^n denotes the space of input vectors. Let ξ be the deviation between $f(x_i)$ and y_i. The optimization problem solved by support vector machine is[17]

$$\begin{aligned} \min_{w,b} \quad & J = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \quad & y_i(w\cdot\Phi(x_i)-b) \geq 1-\xi_i \\ & \xi_i \geq 0, i = 1,\cdots,n \end{aligned} \tag{1}$$

$\Phi(\cdot)$ is the map from input space into feature space, and it is decided by the kernel function $k(x,\hat{x})$. The Lagrangian for this problem is

$$L=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i+\sum_{i=1}^{n}a_i[1-\xi_i-y_i(w\cdot\Phi(x_i)-b)]-\sum_{i=1}^{n}\pi_i\xi_i \tag{2}$$

The Karush-Kuhn-Tucker(KKT) optimal conditions are given by[17]

$$w = \sum_{i=1} a_i y_i \Phi(x_i) \tag{3}$$

$$\sum_i a_i y_i = 0 \tag{4}$$

$$C - a_i - \pi_i = 0, \forall i \tag{5}$$

$$a_i[1-\xi_i - y_i(w\cdot\Phi(x_i)-b)] = 0, \forall i \tag{6}$$

$$\pi_i\xi_i = 0, \forall i \tag{7}$$

$$a_i \geq 0, \pi_i \geq 0, \forall i \tag{8}$$

According to the above KKT optimal conditions, we can obtain

$$\xi_i = \begin{cases} 0, & \text{if } a_i = 0 \\ 0, & \text{if } 0 < a_i < C \\ \geq 0, & \text{if } a_i = C \end{cases} \tag{9}$$

The separating hyper planes and the distribution of training examples in feature space are similar to that in figure 1.

For a training example $\mathbf{x}_i$, if its corresponding Lagrange multiplier a_i is equal to upper bound of *C*, such as $\mathbf{x}_5$, this training example must lie between H_1 and H_{-1} or lie among the training examples of other class. Obviously, this example should be dealt with carefully because it seems like noise or 'dangerous' example that may bring on overlapping. If its corresponding Lagrange multiplier a_i is 0, we can see that it can't contribute on the decision of separating hyper plane from (3), so this example should be excluded from prototype set because the class information contained in it is redundant. If its corresponding Lagrange multiplier $0<a_i<$C, this ex-

ample must lie on the hyper plane H_1 or H_{-1}, and this example plays most important role in deciding the separating hyper plane, so it is top-priority prototype candidate.

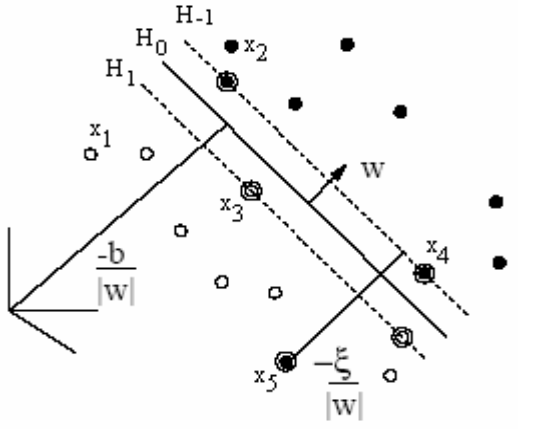

Fig. 1. The separating hyper planes and distribution of training examples in feature space

So, we can find that the support vectors with Lagrange multipliers smaller than *C* are representative examples and contain most classification information and this throws light on developing new prototype selection method. We also observe that the support vectors with Lagrange multipliers equal to *C* are 'dangerous' examples and they should be dealt with carefully. In all, by using the classification information contained in support vectors, we can develop new effective prototype selection methods for nearest neighbor rules.

3 Our Prototype Selection Method Based on Support Vectors

In this section, we will introduce our method according to the introduction in section 2. Suppose that there exists a given training set $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l} \in X^n \times R$. At first, we will choose proper kernel function $k(x, \hat{x})$ and parameter *C*, and then training set *T* is used to learn the SVM. After the SVM is learned, the support vector set noted as *S* is obtained.

We only select those support vectors on the right side near H_0 as prototype candidates, so we defined prototype candidate set P_c as

$$P_c = \{\mathbf{x}_i \mid y_i \times f(\mathbf{x}_i) > 0, \mathbf{x}_i \in S, f(\mathbf{x}_i) \text{ is output of SVM}\} \tag{10}$$

The process to obtain P_c is both condensing and editing process. Deleting non-support vectors is an condensing process, which can condense prototype set effectively, and excluding support vectors lying among examples of other class can avoid

error or overlapping examples' being selected as prototypes. For the SVM performs good generalization ability, and it can select out most representative examples from training set, so prototype selection based on support vectors may obtain better generalization ability than other instance reduction methods.

Prototype candidate set P_c can be used as prototype set, but it may not be the smallest prototype set. In order to obtain minimal set, the condensing process is implemented on P_c. Because there is no noise and overlapping in P_c, simple condensing or deleting process is adequate. Here we use the rule of Drop2 [2] to condense P_c. The rule is

Remove the instance if at least as many of its associates in the original training set would be classified correctly without it.

In Drop2, the distance between two examples should be computed. Because we use the hyper planes of obtained SVM to condense and edit the training set and the hyper planes are linear in feature space, we should use the distance between two examples in feature space. The advantage of selecting prototypes in feature space is obvious.

First, in feature space, the hyper planes are linear. For linear class boundary, fewer points are required to express it than that of nonlinear one, which makes it possible to condense the prototype candidates as small as possible. In another aspect, we can deal with nonlinear and linear SVM with uniform method. The linear SVM can be seen as a special case with kernel function $k(x_i, x_j) = x_i \cdot x_j$.

Let the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in feature space is $d_{ij}^{(H)}$, if the kernel function is $K(\mathbf{x}_i, \mathbf{x}_j)$,we can obtain:

$$\begin{aligned}(d_{ij}^{(H)})^2 &= \left\|\Phi(\mathbf{x_i}) - \Phi(\mathbf{x_j})\right\|^2 \\ &= \left\|\Phi(\mathbf{x_i})\right\|^2 + \left\|\Phi(\mathbf{x_j})\right\|^2 - 2\left\|\Phi(\mathbf{x_i})\right\|\left\|\Phi(\mathbf{x_j})\right\| \\ &= K(\mathbf{x_i}, \mathbf{x_i}) + K(\mathbf{x_j}, \mathbf{x_j}) - 2K(\mathbf{x_i}, \mathbf{x_j})\end{aligned} \tag{11}$$

Because the number of prototype candidates is much smaller than the size of training set *T*, small voting parameter *k* in Drop2 [2] is adequate and large *k* may mistake unrelated examples as neighbors. By applying Drop2 on the P_c, we can obtain the result prototype set P_s.

For an unknown example, we compute the distance between it and all the members of P_s in feature space, and classify it into the class of its nearest neighbor in P_s.

4 Computational Results

In this section, experiments are done to illustrate the performance of our method on 3 benchmark data sets from UCI Repository of machine learning databases [19]. They are Johns Hopkins University Ionosphere database, Wisconsin Breast Cancer Database(WBC) and Wisconsin Diagnostic Breast Cancer(WDBC) database. There are 351 instances described by 34 continuous predictor attributes and one binary class

attribute in Johns Hopkins University Ionosphere database. For Wisconsin Breast Cancer Database, 463 instances are used and they are described by 9 continuous predictor attributes and one binary class attribute. There are 569 instances described by 30 predictor attributes and one binary class attribute in Wisconsin Diagnostic Breast Cancer database.

Experiment 1 is done to illustrate the performance of our method in prototype selection for nearest neighbor rules. It is compared with other popular prototype selection methods Drop4, Drop5 and MCS. Drop4 and Drop5 are editing algorithms; MCS is training-set-consistent condensing algorithm. These popular algorithms and our method are respectively applied to the same data set in order to compare the performance and 10-fold cross validation method is used to obtain average performance. Keerthi's improved SMO algorithm [18] is used to train SVM with training set. The comparison result is shown in table 1.

Table 1. Comparison result between our method and other condensing and editing algorithms

		Ionosphere	WBC	WDBC
Drop4	Reduction rate	8.04%	4.05%	5.39%
	Accuracy	83.43%	92.61%	94.91%
Drop5	Reduction rate	8.23%	6.15%	5.21%
	Accuracy	76.29%	90.87%	93.51%
MCS	Reduction rate	16.36%	14.1%	10.33%
	Accuracy	85.71%	88.91%	93.68%
Our method	Reduction rate	5.47%	2.13%	1.46%
	Accuracy	87.14%	94.57%	95.61%

The comparison result shows that our method obtains higher reduction rate and higher classification accuracy than those of other popular editing and condensing methods. This indicates that our method is superior to current condensing or editing prototype selection method. It also indicates that SVM can help to improve the reduction rate and accuracy when it is used to develop new prototype selection method.

For the SVMs in experiment 1 on three data sets, we list the average number of support vectors of SVM and the average size of result prototype set based on corresponding SVM in table 2.

Table 2. size of support vectors and prototypes

	Ionosphere	WBC	WDBC
Number of support vectors	132	83.9	103.9
Number of prototypes	17.3	8.9	7.9

As we can see in table 2, a small portion of support vectors are selected as prototypes and used to classify new examples. The number of support vectors is much larger than the size of prototype set, and more support vectors will make the decision function of SVM more complex, as a result, the speed in classification phase will be slow, which is substantial for large problems. So our method supplies new method to simplify the classification of SVM.

5 Conclusion

In this paper, SVM is used to select prototypes in order to obtain higher reduction rate and classification accuracy for nearest neighbor rule. Because all the support vectors can decide the classification boundary, so non-support vectors can be excluded from prototype set. As to the support vectors lying among examples of other class, they may result in overlapping and should be excluded from prototype set in order to improve generalization performance.

The training set is used to train a SVM, and then those support vectors on the right side of H_0 in figure 1 will be selected into prototype candidate set. In order to obtain smaller prototype set, the prototype candidate set is condensed with Wilson's Drop2 instance reduction rule to obtain the resulting prototype set. For an unknown example, the distances in feature space between it and all the member of prototype set are computed and it is classified as the class of its nearest neighbor in the prototype set.

Experiment results show that our method is an effective prototype selection method and it can obtain higher reduction rate and classification accuracy than those of popular editing and condensing algorithms. It combines the condensing process and editing process so as to obtain better performance. The comparison between the number of support vectors and the number of prototypes indicates that our method can simplify support vector decision rule, but it should be improved so as to obtain same generalization ability as that of SVM.

References

1. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transaction on Information Theory. Vol. 13, 1(1967) 21-27
2. Wilson, D.R., Martinez, T.R.: Reduction Techniques for Instance-based Learning Algorithm. Machine Learning. Vol. 38, 3(2000) 257-286

3. Toussaint, G.: Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress. Proceedings of INTERFACE-2002, 34th Symposium on Computing and Statistics, Ritz-Carlton Hotel, Montreal, Canada. (2002) 83-106
4. Aha, D.W., Kibler, D., Albert M.K.: Instance-based Learning Algorithms. Machine Learning. Vol. 6, 19(1991) 37-66
5. Dasarathy, B.V.: Minimal Consistent Set (MCS) Identification for Optimal Nearest Neighbor Decision Systems Design. IEEE transaction on System, Man, and Cybernetics, Vol. 24, 3(1994) 511-517
6. Kuncheva, L.I., Bezdek, J. C.: Nearest Prototype Classification: Clustering, Genetic Algorithms, or Random Search. IEEE transactions on System, Man and Cybernetics, 28(1998) 160-164
7. Dasarathy, B.V., Sánchez, J. S., Townsend S.: Nearest Neighbor Editing and Condensing Tools-Synergy Exploitation. Pattern Analysis & Application. 3(2000) 19-30
8. Vapnik, V. N.: The Nature of Statistical Learning Theory. Springer Verlag, New York. (1995)
9. Vapnik, V. N.: Estimation of Dependences Based on Empirical Data. Springer Verlag, Berlin. (1982)
10. Vapnik, V. N., Chervonenkis A.: Theory of Pattern Recognition. Nauka, Moscow. (1974)
11. Kecman, V.: Learning and Soft Computing, Support Vector machines, Neural Networks and Fuzzy Logic Models. The MIT Press, Cambridge, MA (2001)
12. Wang, L.P. (Ed.): Support Vector Machines: Theory and Application. Springer, Berlin Heidelberg New York (2005)
13. Burges C. J. C.: Simplified Support Vector Decision Rules. the 13th international conference on Machine Learning. (1996) 71-77
14. Burges C. J. C., Schölkopf B.: Improving the Accuracy and Speed of Support Vector Machines. In: M. Mozer, M. Jordan, and T. Petsche (eds.): Neural Information Processing Systems, Vol. 9. MIT Press, Cambridge, MA. (1997)
15. Vishwanathan S.V.N., Murthy M. N.: Use of Multi-category Proximal SVM for Data Set Reduction. In: Ajith A., Mario Köppen(Eds.): Hybrid Information Systems. First International Workshop on Hybrid Intelligent Systems, Adelaide, Australia, December 11-12. (2001) 19-24
16. LeCun Y., Jackel L., Bottou L., Brunot A., Cortes C., Denker J., Drucker H., Guyon I., Müller U., Säckinger E., Simard P., Vapnik V.: Comparison of Learning Algorithms for Handwritten Digit Recognition. In: F. Fogelman, P. Gallinari(Eds): Proc. International Conference on Artificial Neural Networks. (1995) 53-60
17. Burges C. J. C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery. Vol. 2, 2(1998) 121-167
18. Keerthi S. S., Shevade S. K., Bhattacharyya C., Murthy K. R. K.: Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation. Vol. 13, (2001) 37-649
19. Blake C., Keogh E., Merz C. J.: UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA. University of California, Department of Information and Computer Science, (1998)