

Pre-extracting Method for SVM Classification Based on the Non-parametric K -NN Rule

Deqiang Han, Chongzhao Han, Yi Yang and Yu Liu
Institute of Integrated Automation,
Xi'an Jiaotong University, China.
deqhan@gmail.com, czhan@mail.xjtu.edu.cn,
jiafeiyi@163.com, aliuyu18@gmail.com

Wentao Mao
MOE Key Lab of Strength and Vibration,
Xi'an Jiaotong University, China.
maowt.mail@gmail.com

Abstract

With the increase of the training set's size, the efficiency of support vector machine (SVM) classifier will be confined. To solve such a problem, a novel pre-extracting method for SVM classification is proposed in this paper. In SVM classification, only support vectors (SVs) have significant influence on the optimization result. We adopt a non-parametric k -NN rule called relative neighborhood graph (RNG) to extract the probable SVs from all the training samples. Experimental results verify that the approach proposed can effectively reduce training set's size and accelerate the learning speed. At the same time, the classification accuracies are still competitive.

1. Introduction

Support vector machine (SVM)[1] is an efficient tool to solve the classification problem. The learning procedure of standard SVM is equivalent to a quadratic programming (QP) problem. The results of SVM classification only depend on support vectors (SVs) in the dataset, which is always a relatively small part of the whole training set. SVM has been widely used in applications such as character recognition, face recognition and regression, etc[2, 3].

Although SVM has many advantages, it still has many problems. In QP problem, sufficient memory is required to store the kernel matrix, which is directly proportional to the size of training set. And the learning time for SVM increases significantly with the increase of the size of training set. Thus SVM is always not efficient for large-scale datasets.

To solve the problems referred, two types of methods were proposed. One type of methods aim to

improve the algorithm of the standard SVM, such as least square SVM (LS-SVM)[4], sequential minimal optimization (SMO)[5] and incremental SVM[6], etc. Another type of methods attempt to make pre-processing or pre-extracting on training set before SVM training to eliminate some unnecessary vectors which are probably not the SVs according to some criteria to reduce size of training sets used, e.g., AGGC[7], SVM-KM[8], CM[9], etc. Although being effective, they still have drawbacks such as eliminating support vectors by mistake and being vulnerable to the noisy data. In some methods[9], the efficiency is also affected by proper selection of parameters for pre-extracting or pre-processing.

In this paper, a novel approach is proposed for pre-extracting SVs based on a non-parametric k -nearest neighbor (k -NN) rule called relative neighborhood graph (RNG). It can effectively reduce the size of training set thus accelerate the learning speed of SVM with little loss of classification accuracy, which can be verified by experimental results provided.

2. Brief introduction of SVM

Linear SVM aims to find an optimal hyper-plane according to criterion of minimum risk. The optimal hyper-plane can maximize the margin. The model of standard SVM is illustrated in (1) and Figure 1.

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i, \quad (1)$$

$$s.t. \quad y_i(x_i^T \mathbf{w} + b) - 1 + \xi_i \geq 0, i = 1, \dots, l;$$

where ξ_i is the slack terms; b is the bias; \mathbf{w} denotes the weight vector, which can determine the optimal hyper-plane; and $C > 0$ is the penalty parameter.

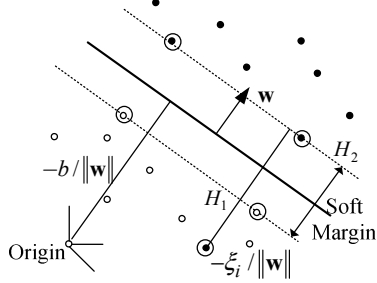


Figure 1. Linear SVM and optimal hyper-plane

The optimization problem described in (1) can be solved by using the Lagrange optimization as follows:

$$\begin{aligned} \max_{\alpha} L_D &= \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \\ \text{s.t.} \quad \sum_{i=1}^l \alpha_i y_i &= 0; 0 \leq \alpha_i \leq C. \quad i=1,2,\dots,l. \end{aligned} \quad (2)$$

where α_i is the Lagrange multipliers, y_i is the class label and \mathbf{x}_i is a sample point. The optimal hyper-plane derived base on (2) is as follows:

$$f(\mathbf{x}) = \text{sgn} \left\{ \sum_{i=1}^n \alpha_i^* y_i (\mathbf{x}_i^T \cdot \mathbf{x}) + b^* \right\} \quad (3)$$

where α_i^* and b^* denote the corresponding parameters of the optimal hyper-plane.

Linear SVM can be generalized to non-linear classifiers through a kernel function:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \quad (4)$$

where $\Phi: \mathbf{x} \rightarrow \Phi(\mathbf{x})$ is the feature mapping from the input space to a usually high dimensional feature space. The corresponding optimal hyper-plane for non-linear SVM is as follows:

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^n \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^* \right) \quad (5)$$

For the optimization problem with constraints, the Karush-Kuhn-Tucker (KKT) condition is very crucial. The KTT condition is described as follows[9].

- 1) $\alpha_i = 0 \Leftrightarrow y_i f_i \geq 1$, ordinary vectors;
- 2) $0 < \alpha_i < C \Leftrightarrow y_i f_i = 1$, SV on the margin;
- 3) $\alpha_i = C \Leftrightarrow y_i f_i \leq 1$, SV between margins.

Based on the conditions listed above, it can be concluded that based on SVM, the original training samples are divided into three parts. The results of SVM classification only depend on SVs in the dataset. If we could pick out the samples, which are probable support vectors before the training procedure, the learning time can be reduced and the classification

accuracies can be maintained. Based on such an idea, we propose a method to pre-extract training samples based on a non-parametric k -NN called relative neighborhood graph (RNG)[10], where parameter k can be determined automatically based on geometric relationship defined. RNG is introduced in next section.

3. A non-parametric k -NN rule – relative neighborhood graph

Neighborhood-based rules, such as nearest neighbor (NN) and k -NN, graph neighbor (GN)[10], etc., are always effective in pattern classification. Based on k -NN rule, the neighborhood information of corresponding sample can be derived, which is useful for sample selection or pre-extraction. The main problem for k -NN is the determination of optimal value of k . Relative neighborhood graph (RNG) rule, which is a kind of GN, can be regarded as a non-parametric k -NN, the value of k can be determined automatically.

\mathbf{x}_q is a query sample. If the following relationship:

$$\begin{aligned} d(\mathbf{x}_q, \mathbf{x}_i) &\leq \max(d(\mathbf{x}_q, \mathbf{x}_j), d(\mathbf{x}_i, \mathbf{x}_j)) \\ \forall \mathbf{x}_j \in X, i \neq j \end{aligned} \quad (6)$$

comes into existence, \mathbf{x}_i is called a relative neighborhood graph neighbor (RNGN) of \mathbf{x}_q . X is the whole dataset. $d(\cdot, \cdot)$ represents the distance. Euclidean distance is always used.

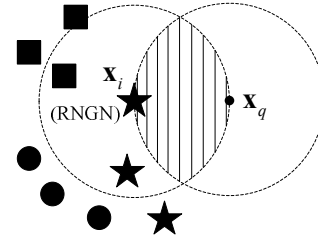


Figure 2. Relative neighborhood graph neighbors

As illustrated in Figure 2, geometric interpretation of RNG is based on the concept of lune, defined as the intersection between two hyper-spheres centered at \mathbf{x}_i and \mathbf{x}_q and whose radius are the distance between \mathbf{x}_i and \mathbf{x}_q . If there is no other training point lies in the lune defined, \mathbf{x}_i is called the RNGN of \mathbf{x}_q . For a sample \mathbf{x}_q , we can find all its RNGNs. If there exist RNGNs belonging to the class, which is not the same as the class of \mathbf{x}_q , \mathbf{x}_q and such RNGNs are likely to be near the boundaries among different classes. That is to

say based on the neighborhood information indicated by RNGNs, we can judge the rough location of the corresponding sample (whether it is near the boundaries among classes or not). And then we can make the pre-extracting operation on the dataset. The specific method is introduced in the next section.

4. Pre-extracting method for SVM classification based on RGN rule

For two-class (or binary) classification problem, if there exist a sample's RNGNs belonging its composite class, then the corresponding RNGNs is likely to be near the boundaries between the two classes. The samples surrounding the boundaries are probable to be the SVs. Suppose that the two classes are class 0 and class 1. Pre-extraction on the training set is as follows:

- 1) Divide the original training set T into set T_0 and T_1 , which include the training samples belonging to Class 0 and Class 1 respectively.
- 2) For each sample in T_0 , find its corresponding RNGNs. Tag the RNs belonging to Class 1. All the samples tagged compose a set S_0 .
- 3) For each sample in T_1 , find its corresponding RNs. Tag the RNGNs belonging to Class 0. All the samples tagged compose a set S_1 .
- 4) Let $T' = S_0 \cup S_1$, T' is the pre-extracted training set for SVM training procedure.

For non-linear SVM, the definition for distance should be modified as follows:

$$d^\Phi(\mathbf{x}_i, \mathbf{x}_j) = \|\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\|_2 \quad (7)$$

$$= \sqrt{K(\mathbf{x}_i, \mathbf{x}_i) - 2K(\mathbf{x}_i, \mathbf{x}_j) + K(\mathbf{x}_j, \mathbf{x}_j)}$$

where $K(\cdot, \cdot)$ denotes the kernel function.

5. Experiments

In the experiments, some artificial datasets and real-world datasets are used. All the experiments are carried on a PC with CPU: AMD Athlon 4000+ Dual Core, 2.11GHz; RAM: 2GB DDR II; Operating System: Windows XP -SP2; Software Compiler: Matlab R2007a. SVM model used is LS-SVM[4]. Experimental results of pure LS-SVM and LS-SVM with pre-extracting method are compared to verify the performance of the pre-extracting method proposed.

5.1. Two-spiral dataset

The two-spiral is a classical two-class classification problem. Two-spiral can be generated as follows[11]:

$$spiral1: \begin{cases} x = A\theta \cos(\theta) \\ y = A\theta \sin(\theta) \end{cases} \quad (8)$$

$$spiral2: \begin{cases} x = A\theta \cos(\theta + \pi) \\ y = A\theta \sin(\theta + \pi) \end{cases} \quad (9)$$

In the experiment, $A = 3$, $\pi/2 \leq \theta \leq 3\pi$. Suppose that each class's probability density is uniform along the corresponding curve. The two-spiral dataset used is polluted by Gaussian noise whose mean is 0 and variance is 1.5. Pure and polluted two-spiral curves are shown in Figure 3. Totally 500 samples are generated. Class 0 and Class 1 each has 250 samples. Randomly select 125 training samples from each class and the remainders are reserved for test. The classification procedure referred is executed for 10 times.

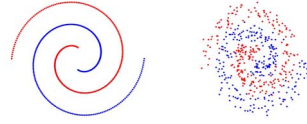


Figure 3. Pure and polluted two-spiral curves

The kernel function adopted is the radial basis function (RBF) as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\| / \sigma^2) \quad (10)$$

The parameter is appropriately selected as follows: penalty parameter is 2.5 and σ^2 in (10) is set to 0.5. The experimental results are listed in Table 1. #SV represents the quantity of SV.

Table 1. Experimental results on 2-spiral (Average)

Method	Accuracy	#SV	Training time (Sec)
No pre-extracting	97.04%	179.5	0.2564
With pre-extracting	96.32%	59.1	0.0269

5.2. Ripley dataset

The Ripley dataset[12] has of two classes. The data for each class have been generated by a mixture of two Gaussian distributions. There are 250 training samples (125 per class) and 1000 test samples (500 per class).

Linear kernel function is adopted and penalty parameter is appropriately selected to be 2. The experimental results are listed in Table 2.

Table 2. Experimental results on Ripley

Method	Accuracy	#SV	Training time(Sec)
No pre-extracting	89.20%	175	0.0178
With pre-extracting	88.20%	42	0.0029

5.3. Pima Indians diabetes dataset

Pima Indians diabetes dataset (2-class) is from UCI[13] dataset. Class 0 has 500 samples and Class 1 has 258 samples.

Standardize all the attribute values of all samples before the experiment.

Linear kernel function is used and penalty parameter is appropriately selected to be 5. Experimental results are listed in Table 3.

Table 3. Experimental results on Pima (Average)

Method	Accuracy	#SV	Training time(Sec)
No pre-extracting	76.61%	177.7	0.0580
With pre-extracting	75.32%	99.70	0.0128

Based on the experimental results listed above, it can be concluded that according to the pre-extracting method proposed for SVM classification, the learning speed can be improved significantly and the corresponding classification performance can still be competitive.

6. Conclusions

In this paper, to deal with the problem of SVM learning in large-scale dataset, a novel pre-extracting approach is proposed. The experimental results provided can verify the efficacy of the approach proposed. The time cost for SVs' pre-extracting based on the method proposed is large. The averaged cost to search the RNGNs of a sample is close to $O(dn)$, for a training set with n samples in d -dimension feature space. But it is accomplished offline.

It should be noted that the approach proposed can not counteract the negative effect of noisy data or isolated data. And the approach proposed is according to two-class classification problems. Pre-extracting approach for multi-class problem is worth researching. There are still lots of research works for us to do in the future.

Acknowledgements

This work is supported in part by National Natural Science Foundation of China: No.60574033 and No.60602026, in part by Grant for State Key Program for Basic Research of China (973) No.2007CB31006

References

- [1] V. N. Vapnik. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [2] K. R. Mülle, S. Mika, G. Ratsch, K. Tsuda, and B. Scholchopf. An introduction to kernel-based learning algorithm. *IEEE Transactions on Neural Networks* 12(2), pp. 181-201, 2001.
- [3] A. Smola and B. Scholchopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica* 22(1), pp. 211-231, 1998.
- [4] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Process Letter* 9(3), pp. 293-300, 1999.
- [5] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In: *Advances in Kernel Methods-Support Vector Learning*, 1998, pp. 185- 208.
- [6] Z. W. Li, J. Yang, and J. P. Zhang. Dynamic Incremental SVM learning Algorithm for Mining Data Streams. In: *The First International Symposium on Data, Privacy, and E-Commerce*, 2007, pp. 35-37.
- [7] I. D. Guedalia. An on-line agglomerative clustering method for non-Stationary data. *Neural Computation* 11, pp. 521-540, 1999.
- [8] M. B. Almeida, A. P. Braga, and J. P. Braga. SVM-KM: speeding SVMs learning with a priori cluster selection and k-means. In: *Proceedings of the sixth Brazilian Symposium on Neural Networks*, Los Alamitos, 2000, pp. 162-167.
- [9] D. Y. Meng, Z. B. Xu, and W. F. Jing. A more efficient preprocessing method for support vector classification. In: *Proceedings of International Conference on Neural Networks and Brain*, Beijing, 2005, pp. 1173-1177.
- [10] J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE* 80(9), pp. 1502-1517, 1992.
- [11] H. Du and Y. Q. Chen. Rectified nearest feature line segment for pattern classification. *Pattern Recognition* 40(5), pp. 1486-1497, 2007.
- [12] B. D. Ripley. *Pattern recognition and neural Networks*. Cambridge: Cambridge University Press, 1996.
- [13] C. L. BLAKE and C. L. Merz, *UCI repository of machine learning databases*, <http://www.ics.uci.edu/~mllearn/MLRepository.htm>