Editing Support Vector Machines

Haixin Ke and Xuegong Zhang

Dept of Automation, Tsinghua University / State Key Lab of Intelligent Technology and Systems
Beijing 100084, China
hxke@simba.au.tsinghua.edu.cn, zhangxg@tsinghua.edu.cn

Abstract

A support vector machine constructs an optimal hyperplane from a small set of samples near the boundary. This makes it sensitive to these specific samples and tends to result in machines either too complex with poor generalization ability or too imprecise with high training error, depending on the kernel parameters. In this paper, we present an improved version of the method, called editing support vector machine or ESVM, which removes some samples near the boundary from the training set. Experiments show that for cases that the two classes are overlapped, ESVM can get better generalizing ability, and ESVM is also more robust with noises.

I. INTRODUCTION

The support vector machine or SVM paradigm, which was developed from the theory of structural risk minimization and was first introduced by Vapnik and his colleagues, is a new pattern recognition technique that aims at better generalizing ability with limited training samples [1]. Using the data near the boundary regions between the two classes, SVM constructs an optimal separating hyperplane with maximum margin between the two classes. This can be formulated as in the follow way [2]:

Given a set of training samples $(\boldsymbol{x}_i, y_i)$, $i = 1,...,n$, $\boldsymbol{x} \in R^d$, $y \in \{+1,-1\}$, find a hyperplane $\boldsymbol{x} \cdot \boldsymbol{w} + b = 0$ with the minimum value of $\frac{1}{2}\|\boldsymbol{w}\|^2$ under the constraints of

$$y_i[(\boldsymbol{w} \cdot \boldsymbol{x}_i) + b] - 1 \geq 0, \quad i = 1, \cdots, n.$$

SVM can be generated to nonlinear cases by mapping the input vectors into a high dimensional space, using a proper kernel function. As to nonseparable cases, a special penalty $C\sum_{i=1}^{n}\xi_i$, which measures the amount of violation to the constraints, is introduced in the objective function, where C is a constant controlling the penalty on the misclassified samples.

Because of its many merits, SVM has been applied to a variety of problems. However, it also leaves much in debate. As noted in [1] and [3], the support vectors or SVs, which are those closest to the optimal hyperplane, play a crucial role in constructing the decision function. When the two classes overlap, the samples in the overlapped region are always nearest to the separating hyperplane, so they are given serious consideration during training. If the parameters of the kernels and the C value are not properly chosen, these samples may either cause the decision boundary to be too complicated with poor generalizing ability, or lead to a result that has high training error rate. Moreover, if there exists some noises in this region, their effects would be overstated.

To overcome this problem, an intuitive idea is to eliminate those samples in the overlapped area. Following this consideration, we proposed an improved version of SVM called Editing Support Vector Machines or ESVM. Its basic scheme is similar to that of editing nearest neighbor methods in statistical pattern recognition, which randomly divides the training set into subsets, using one subset to edit the other one and then get the final decision boundary using the remained samples (details can be found in several textbooks on pattern recognition, such as [4]). Here we do not perform the partition phase, but apply editing on the training set directly. This procedure eliminates the samples in the overlapped region so that a clearer boundary between the two classes can be obtained, with better generalization ability. Experiments show that the proposed new method is quite promising, especially for overlapped cases and for training sets corrupted with noises and outliers.

II. THE ESVM SCHEME

Conceptually, our purpose is to remove the questionable samples in the overlapped region, or to be except, get rid of those incorrectly classified samples, for the existence of these error samples can only cause bad effect on get-ting the decision function. Figure 1 illustrates the idea in one-dimensional case. There is an overlap in the distributions of the two classes. The effect of our editing procedure is to remove those training samples in the overlapped region as illustrated by the dark areas in the distributions.

This work was supported by the National Natural Science Foundation of China, under project number 69885004.

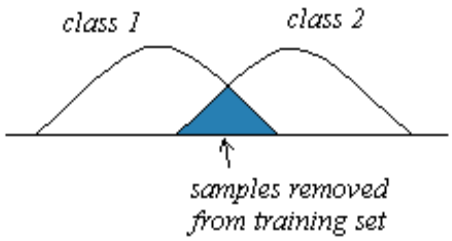

Figure 1. The basic idea of ESVM

However, it is hard to specify which sample would be misclassified before the correct decision function is found. According to the setting of the problem, all we have for the task is the training samples. So we need to train a preliminary machine using the samples first, and then use this trained machine as a criteria for judging which sample should be removed.

Consider the training set

$$(\boldsymbol{x}_i, y_i),\ i = 1,...,n,\ \boldsymbol{x} \in R^d,\ y \in \{+1,-1\}$$

The dual optimization problem of SVM is: maximize the quadratic objective function

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

subject to constraints

$$\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$0 \le \alpha_i \le C,\quad i = 1, \cdots, n$$

where $K(\boldsymbol{x}_j, \boldsymbol{x}_j)$ is the kernel function, and α_i are the Lagrange multipliers corresponding to each training samples, and C is the penalty constant.

Solving this optimization problem according to the original training set, we obtain an optimal hyperplane that separates the two classes with some nonseparable samples. We call this hyperplane the original hyperplane. And the decision function would be:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*$$

where α_i^*, b^* are the optimal coefficients.

From the derivation of the SVM algorithm, we know that the absolute function values for the SVs are 1 or -1, and those for the other correctly classified samples are either larger than 1 or less than -1. The samples that lie on the wrong side of the hyperplane can be easily identified according to its output value. We can remove these samples from the training set and use the remained set to train the SVM again. The effect of this procedure can be seen in the example of Figure 2.

Removing only these nonseparable samples can modify the decision hyperplane, but the effect is not notable, since the new SVs always remain the same as the original ones. To get better improvement, we introduce a gate value g, which is a predetermined threshold. If we assign the value of g to be slightly greater than 1, and remove all those samples whose absolute function values are less than g, as well as those misclassified samples, a new training set will be obtained in which there is an isolated area or "clear zone" between the two classes. Training SVM with this edited new training set will result in a smoother separating hyperplane, which we call edited hyperplane.

This edited hyperplane may be less accurate since the samples nearest to the optimal decision boundary are excluded, including all the previous support vectors. However, it can still reflect the major structure of the two classes, ignoring some less important details.

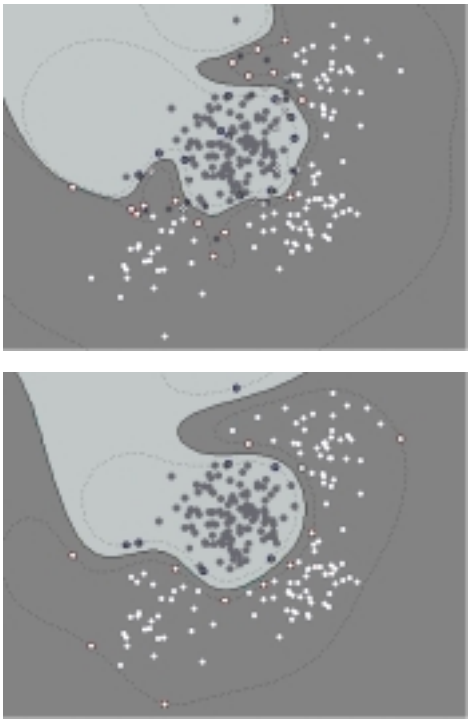

Figure 2. The SVM decision hyperplanes with the nonseparable samples in the training set (upper) and without these nonseparable samples (lower).

If it is necessary, we can get some omitted details back by taking one further step. Using the edited hyperplane to edit the original training set, and this time removing only

the misclassified samples, we then obtain a new version of edited training set. With this training set, SVM can get a better decision boundary that is free from the bad effect of the error samples and yet contains rich detailed information.

We call the above scheme as Editing SVM or ESVM. Different implementations can be designed according to different situations.

III. Experiment Examples

We generated some artificial data to test the proposed ESVM scheme. In these experiments, the distribution of each class is mixture Gaussian, and the kernel in SVM is radial basis function (RBF). In all the experiments, and in the different phases of our ESVM procedure, we adopted the same width parameter for the RBF kernel, and the same penalty constant C, so that the results can be comparable.

In order to compare the performance of the original SVM and ESVM, we randomly divide the samples into two sets (of the same size), one for training, and one for testing.

Figure 3 and figure 4 are two examples of our experiments. From these experiments, we can see that because of the existence of those misclassified samples, the results of standard SVM tend to be too sinuous, or one can say that the resulted machines are too complex to have good generalizing ability. On the other hand, the decision boundaries obtained by ESVM is much smoother (implying better generalizing ability) and yet the crucial classification information is well represented.

IV. Conclusions and Further Discussion

Although the scheme of ESVM still deserves further theoretical study, we observed from experiments that it is promising. By removing those questionable samples, the machine tends to be simpler and thus have better generalizing ability. In statistical learning theory, an important result is that the upper bound of the expected risk contains two parts, one is the empirical risk, and the other is the so-called confidence interval, which is decided by the VC dimension of the learning machines and the size of the training set [1]. Here from our study, we tend to believe that the confidence interval could also be affected by the distribution style of the training data. If this can be proven theoretically, then the work of ESVM is to edit the training set so that it will have a distribution that is favorable for better generalizing ability.

On the other hand, by applying the editing procedure, the machine take some non-support vectors into account, so that it tends to be less sensitive to some specific samples (such as the outliers studied in [3]). In this sense, ESVM is more robust with noises.

ESVM may also have some contribution to the choice of kernel and its parameters, which is very essential for SVM. If the kernel is not suitable, or the parameters of the kernel is not suitable, the result may become too complicated and have less generalizing ability, as seen in the examples presented in the previous section. However, this problem does not exist for ESVM (or at least not as severe as in the standard SVM case). ESVM can construct a relatively simple decision hyperplane using samples distributed in a rather complex manner. This shows that the performance of ESVM does not heavily depend on the choice of kernel parameters. In fact, in our experiments, we tried very small width parameter for the RBF kernel (for which standard SVM will result in rather poor result) and still get quite good results.

The computation labor of ESVM should also be considered. Getting the original hyperplane using standard SVM is time-consuming, especially when the overlapped area is large. However, the edited hyperplane is much easier to get, since there is already a clear zone between the two classes of samples. And in the final phase, we can use the edited hyperplane as a good initial value for the algorithm, which can save much computing. So the bottleneck lies still in the preliminary training phase. It may be a good idea to use some other classification method but not SVM to get the preliminary hyperplane. The feasibility of this idea is still to be studied.

References

[1] V.N. Vapnik, *Statistical Learning Theory*, 2nd ed., NY: Springer-Verlag, 1999

[2] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, vol.2 no.2, 1998, pp.1-43

[3] Xuegong Zhang, "Using class-center vectors to build support vector machines", *Neural Networks for Signal Processing* IX, 1999, pp.3-11

[4] Z. Bian and X. Zhang, *Pattern Recognition*, 2nd ed., Beijing: Tsinghua University Press, 2000

[5] X. Zhang and H. Ke, "ALL/AML cancer classification by gene expression data using SVM and CSVM approach", A. Keith Dunker, et al.(eds.) *Genome Informatics 2000*. Tokyo: Universal Academy Press, 2000, pp. 237-239

[6] X. Zhang and H. Ke, "Central support vector machines and its application to cancer classification", submitted to *IEEE Trans. on Neural Networks*

[7] Steve Gunn, *Support Vector Machines for Classification and Regression*, ISIS Technical Report, University of Southampton, 1998

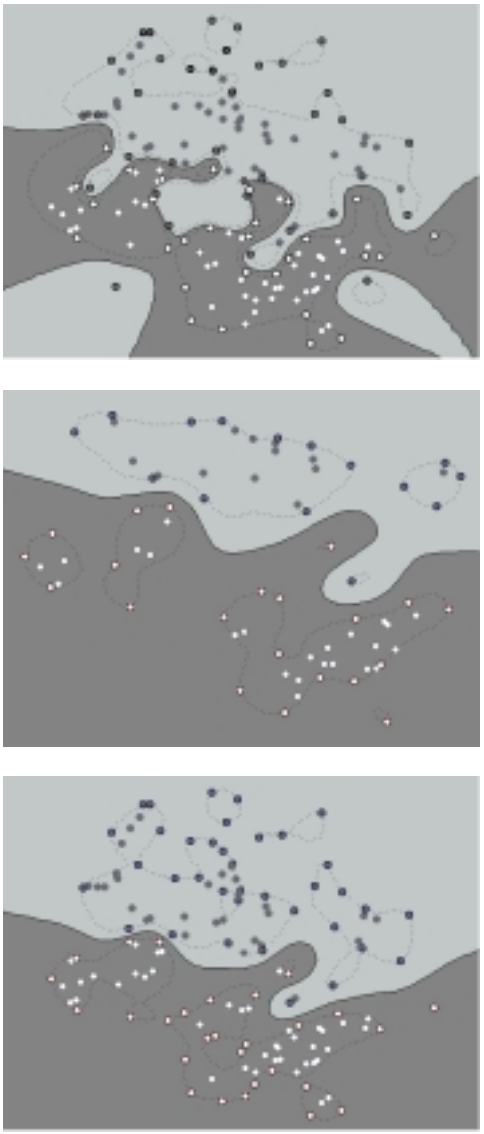

Figure 3. Experiment of ESVM. The decision boundaries obtained by standard SVM (top), edited hyperplane (middle) and by ESVM (bottom). (RBF width σ =0.3, C=10000, g=1.1)

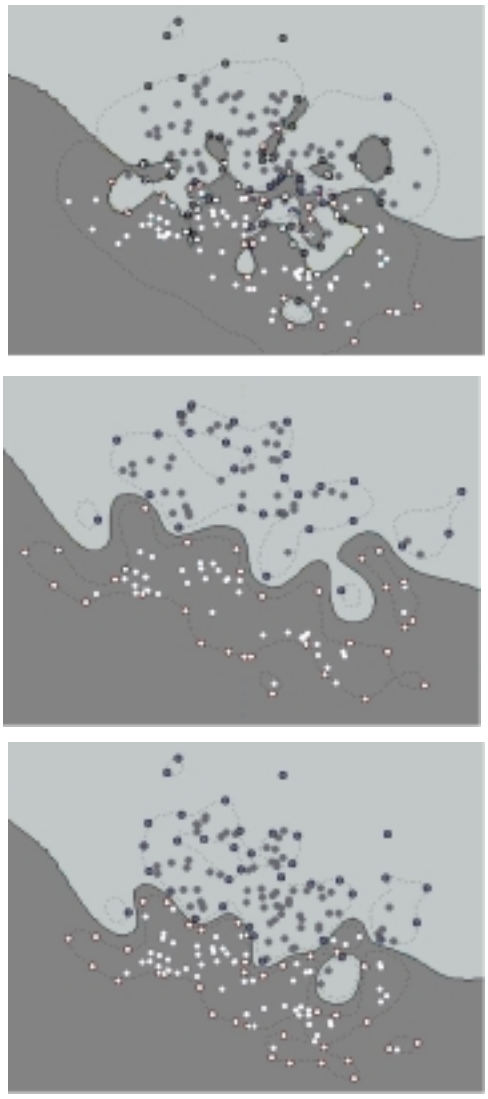

Figure 4. Experiment of ESVM. Top: hyperplane obtained by SVM, test error 0.268; Middle: the edited hyperplane, test error 0.171; Bottom: hyperplane obtained by ESVM, test error 0.167. (RBF width σ =0.3, C=10000, and g=1.01)