SVM-Based Classifier Design with Controlled Confidence

Mingkun Li, Ishwar K. Sethi
Intelligent Information Engineering Lab
Department of Computer Science and Engineering
Oakland University
Rochester, MI 48309
{li, isethi}@oakland.edu

Abstract

A new classification methodology with controlled error rates and a reject option is proposed in this paper. The proposed methodology is implemented using Support Vector Machine’s (SVM’s) posterior probability preserving property. A new nonparametric method is proposed to accurately estimate error rates from the output of a trained SVM. The experimental results clearly demonstrate the efficacy of the suggested classifier design methodology.

1. Introduction

When a classifier withholds its decision on an input, it is said to reject the input. Depending upon an application, the rejected input pattern might undergo a manual classification or the input might be captured again. The reject option is generally used in a classifier to safeguard against excessive misclassification or to control error rates. Despite the tremendous work in pattern recognition of last 40 years, no systematic methodology for designing a classifier with controlled error rates with a reject option exists in the literature [1, 7]; the reject option in practice is exercised through empirical means by controlling some threshold setting.

This paper proposes a new classification methodology with controlled confidence. The user is allowed to specify error rates for this classifier. If the classifier cannot predict any test sample with the specified error rates, it rejects it. The proposed classification methodology is implemented using the support vector machine (SVM). Since the output of a SVM is a distance from the separating hyperplane and not a probability, it cannot be directly related to error rates. A new nonparametric approach is therefore proposed to accurately estimate the error rates from a trained SVM. Even though this approach cannot guarantee the optimality of the resulting classifier, it should give sub-optimal solution for the properly trained SVMs.

The two-class classification problem is used to describe the methodology, although it can also be easily applied to multiple-class cases. The organization of the paper is as follows. Section 2 presents the optimal classifier design with controlled confidence. Section 3 first describes a new method to map the output of a trained SVM to error rates; it then gives an algorithm for training the controlled classifier. Section 4 presents experimental results to demonstrate the proposed methodology. Finally, we conclude with a summary in Section 5.

2. Optimal classifier with controlled Confidence

Suppose we want to classify a feature vector \( x \) into two classes, \( \omega_1 \) and \( \omega_2 \), and we know the posterior probabilities, \( p(\omega_1|x) \) and \( p(\omega_2|x) \). Chow [4] gave the following optimal classification and rejection rule:

\[
\begin{align*}
\text{Decide } & \omega_1 \text{ if } p(\omega_1|x) \geq (1-t) \\
\text{Decide } & \omega_2 \text{ if } p(\omega_2|x) \geq (1-t) \\
\text{Otherwise, reject } & x
\end{align*}
\]

Here \( t \) is a constant between 0 and 0.5, specified by the user. Chow also proved that \( t \) is the upper bound of the error rate.

Chow’s theory paves a way for a classification system with a reject option. However, obtaining reliable and accurate estimates of the posterior probabilities from a limited number of samples is very difficult. Even though the conditional error for a given feature vector is important for some applications, in most cases error rates for different classes are our main concern. This suggests that we may control error rates instead of conditional errors. By controlling error rates instead of conditional errors, we relax the strict criteria and may be able to get better results.

Suppose a classifier partitions the feature space into three regions, \( R_1, R_2, \) and \( R_3 \), where all the vectors in \( R_1 \) and \( R_2 \) are classified into class \( \omega_1 \) and \( \omega_2 \), respectively, and all the vectors in \( R_3 \) are rejected. Then the respective error rates for class \( \omega_1 \) and \( \omega_2 \) are
Given $E_1$ and $E_2$, design a classifier with $R$ as small as possible.

It is obvious that Chow’s classifier is optimal if we choose the smallest $t$ which satisfies the specified error rates. We would, however, face the difficult task of estimating the posterior probabilities. On the other hand, we can make the classifier design task relatively easier if we can transform the feature space into a lower dimensional space where the probability estimation or error rates estimation is easier, and the transformation preserves the order of probabilities. The resulting classifier should be optimal or sub-optimal. For a properly trained classifier, it should maintain the order of the probabilities [5, 6]. Thus, though we can’t guarantee the optimality of resulting classifier, we expect it to give reasonable results. The next section proposes an implementation for a classifier with controlled error rates along these lines using SVMs.

3. SVM implementation with controlled error rates

To implement the optimal classifier, we use the SVM as our base classifier for its strong theoretical foundation and superior performances in many real-world applications [2, 3]. Furthermore, the output of a SVM is in the same order of posterior probabilities [5]. Basically, the SVM maps data samples onto a higher dimensional space in which they are linearly separable; then the SVM finds a maximum margin hyperplane to partition the samples. By using the maximum margin principle the SVM technique guarantees good generalization performance. However, an SVM output is not a probability but a distance from the separating hyperplane. Thus, we need a way to transform SVM output scores to probabilities; or we need a method to reliably and accurately estimate error rates from a trained SVM.

Let the output of a trained SVM be:

$$f(x) = \sum_i \alpha_i y_i K(s_i, x) + b$$

where $K$ is a kernel function and $s_i$’s are support vectors. Platt [5] suggests using a sigmoidal function to transform SVM outputs to posterior probabilities such as:

$$P(y = \omega_i | f) = \frac{1}{1 + \exp(A_i f + B_i)}, \ i = 1, 2$$

However, there is no theoretical proof that a posterior probability has sigmoidal shape and most likely it does not. For example, if the distributions of two classes follow Gaussian distributions with different variances, it is obvious that the posterior probabilities do not have sigmoidal shapes. Another obvious solution is to use the histogram of output scores. Here we are faced with the problem of choosing the number of bins [6]. The main problem lies in how to estimate probabilities from limited samples.

However, since error rates are the only concern for the optimal classifier, there is no need to estimate probabilities. We can directly estimate error rates from training samples. Our method is to work on the histograms of the output scores in the training set. Most importantly, we concentrate on the output scores of misclassified support vectors and take them as the bin locations. The informal reasoning is that misclassified support vectors are the most difficult classified samples and thus they convey more information about the underlying distributions.

For a trained SVM and a given training set $T$, we first define two metrics to approximate error rates, $Error^{-1}(f_{-i})$ and $Error^{+1}(f_{+i})$ as:

$$Error^{-1}(f_{-i}) = \frac{\text{card}(x| f(x) \leq f_{-i}, \text{cls}(x) = \omega_{-1}, x \in T)}{\text{card}(x| f(x) \leq f_{-i}, x \in T)}$$

$$Error^{+1}(f_{+i}) = \frac{\text{card}(x| f(x) \geq f_{+i}, \text{cls}(x) = \omega_{+1}, x \in T)}{\text{card}(x| f(x) \geq f_{+i}, x \in T)}$$

where $\text{card}$ is the cardinality of a set and $\text{cls}$ provides the class memberships of samples. We argue that these two error metrics approximate well the error rates for a properly trained SVM. A properly trained SVM should rank samples from the most negative samples to the most positive samples. That is to say, the SVM preserves the order of posterior probabilities. We will illustrate this in our experimental results.

The training algorithm of the optimal classifier is given in Figure 1 for given error rates of two classes. The first step is to train a SVM using training data. The SVM must be properly trained. It cannot be over-trained to cause over fitting. Cross-validation can be used to avoid over fitting. Then we rank the output scores of the misclassified support vectors and search a number called NegativeBound and a number called PositiveBound from the scores. Basically, the NegativeBound is the highest score whose $Error^{-1}$ is less than the given error rate for the negative class and the PositiveBound is the lowest score whose $Error^{+1}$ is less than the given error rate for the positive class.
The classification phase of the optimal classifier is pretty simple. Any sample whose output score from the SVM is less than the NegativeBound is classified as negative sample; any sample whose output score from the SVM is larger than the PositiveBound is classified as positive sample; all other samples are rejected because the decision is too risky.

Algorithm: training of an optimal classifier with controlled error rates using SVM

Input
- X: training samples
- T: class labels for X
- ErrorNeg: error rate for the negative class
- ErrorPos: error rate for the positive class

Output
- NegativeBound: the upper bound for the negative class
- PositiveBound: the lower bound for the positive class

Procedure
1. Properly train a SVM using X and T
2. Find all the support vectors whose class is positive and whose score is negative, denoted as SVPN
3. Rank in ascending order the scores of SVPN, denoted as S_SVPN, and its size as NN
4. S_SVPN(NN+1) = 0, i=1
5. Do
   - NegativeBound = S_SVPN(i)
   - Compute Error\(^{-1}\) of S_SVPN(i)
   - i = i + 1
   While (Error\(^{-1}\) <= ErrorNeg) and (i <= NN+1)
6. If i <= NN+1, NegativeBound = S_SVPN(i-2)
7. Perform similar steps from 2 to 6 to find the PositiveBound

Figure 1. Algorithm to train the optimal classifier with controlled error rates

4. Experimental Results

Two experiments are discussed here. The first one is a synthesized example with known underlying distributions; it is used to demonstrate the posterior probability preserving property of the SVM, which is the foundation of our optimal classifier algorithm with a reject option and controlled error rates. The second experiment demonstrates the basic idea and the advantages of the new algorithm on the widely used UCI Adult benchmark dataset predicting income levels based on census data.

4.1. Synthesized example

We first set up experiments with known distributions to demonstrate the posterior probability preserving property of trained SVMs. 200 2-dimensional samples are drawn from each of two Gaussian distributed classes. The negative class has mean (0, 0), variance (1, 1), and covariance 0; the positive class has mean (2, 2), variance (1, 1), and covariance 0. The samples are shown in Figure 2 and true statistics of the training examples are shown in Table 1.

From the Bayesian decision theory and simple mathematical derivation, we know that the optimal decision boundary is \( f(x_1, x_2) = x_1 + x_2 - 2 = 0 \), and that the posterior probabilities are the same along the contour of \( f(x_1, x_2) \).

![Figure 2. SVM results of two Gaussian classes](image)

We use these 400 samples to train a linear SVM. The results are shown in Figure 2. The negative samples are plus dots; the positive samples are cross dots; the points circled are support vectors, and the black line is the decision boundary. The learned object function is \( g(x_1, x_2) = 0.60x_1 + 0.61x_2 - 1.29 \), which is very close to the theoretical one by a scaling factor. This result clearly demonstrates the posterior probability preserving property of the SVM. It also shows the applicability of SVMs to the optimal classifier.

4.2. UCI Adult census dataset

The UCI Adult database [8] is a well-known and widely used database for pattern classification and data mining tasks. There are 16 prediction results for the UCI database in the UCI website and many others can be found from literatures. Basically, the common error rate is around 15% with the best being 14.05%. The purpose of
our experiments is not to show an improvement in classification rate using our algorithm, which, we believe, is impossible; instead, we use it to show how to control error rates and to illustrate the concept of reject.

The Adult database has 14 attributes such as age, work class, education, occupation, race, native country, and so on. The task is to predict if a household income is greater than 50K per year or not. The Adult database has 30,162 samples as training set and 15,060 samples as test set. Currently we use only first 6,032 samples from the training set to train a SVM and all of the 15,060 test set samples to test.

The SVMlight [9] was used to train the SVM. We used the linear kernel. For the SVM, the error rate is 15.90% for the test set, which is comparable with other algorithms. The results of our optimal classifier are shown in Table 2 for different error settings for two classes. The first two columns show the error rate settings; the next three columns show the error rates obtained. The reject column shows the percentage of rejected samples. From this table, we can see that the test error rates do follow the specified error rates. Another direct observation is that the reject ratio increases with decreasing error rates, which conforms to our intuition. From these results, we can see that our optimal classifier can avoid high-risk decisions and improve the reliability of the classification.

Table 2. Experimental results for adult database

<table>
<thead>
<tr>
<th>Settings</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Neg.</td>
</tr>
<tr>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>7.5%</td>
<td>7.5%</td>
</tr>
<tr>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>10%</td>
<td>10%</td>
</tr>
</tbody>
</table>

To understand the underlying principle of our optimal classifier methodology, we plot the histograms of the output scores for both training samples and test samples in Figure 3. These histograms clearly show that the classifier has a better chance to make a correct decision for the samples the output score of which is far away from the 0, the traditional SVM classification boundary, and that it is very likely to make a mistake for the samples with score near 0. We should avoid the decisions at the boundary regions, and reject these decisions. For example, the error rate for rejected samples is 38.1% for the experiment with the specified error rate 10% for both positive and negative class (last row in Table 2); if we use the dominant class as predicted class, the error rate is 41.5%.

From Table 2, we can see that the test error rates are not exactly the same as the specified error rates. This may have resulted from incomplete SVM training, or error rate estimation methods. This phenomenon demonstrates the difficulties of pattern classification problems. Our algorithm is not optimal, but it does provide a sub optimal solution.

5. Conclusion

A new classification methodology is proposed in this paper on how to design a classifier with controlled error rates and a reject option. It is implemented using the SVM’s posterior probability preserving property. A new nonparametric method was proposed to accurately estimate error rates from the output of SVMs. The experimental results clearly demonstrate the idea and the efficacy of the new classifier.

6. References