Pattern Classification Using Support Vector Machine Ensemble

Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, Sung Yang Bang
Department of Computer Science and Engineering
Pohang University of Science and Technology
San 31, Hyoja-Dong, Nam-Gu, Pohang, 790-784, KOREA
{grass,snpang,invu71,dkim,sybang}@postech.ac.kr

Abstract

While the support vector machine (SVM) can provide a good generalization performance, the classification result of the SVM is often far from the theoretically expected level in practical implementation because they are based on approximated algorithms due to the high complexity of time and space. To improve the limited classification performance of the real SVM, we propose to use an SVM ensemble with bagging (bootstrap aggregating) or boosting. In bagging, each individual SVM is trained independently, using randomly chosen training samples via a bootstrap technique. In boosting, each individual SVM is trained using training samples chosen according to the sample's probability distribution, which is updated in proportion to the degree of error of the sample. In both bagging and boosting, the trained individual SVMs are aggregated to make a collective decision in several ways, such as majority voting, LSE(least squares estimation)-based weighting, and double-layer hierarchical combining. Various simulation results for hand-written digit recognition and fraud detection show that the proposed SVM ensemble with bagging or boosting greatly outperforms a single SVM in terms of classification accuracy.

1. Introduction

The support vector machine is a new and promising classification and regression technique proposed by Vapnik and his group at AT&T Bell Laboratories [4]. The SVM learns a separating hyperplane to maximize the margin and to produce a good generalization capability [3]. Recent theoretical research work has solved existing difficulties in using the SVM in practical applications [8]. Until now, it has been successfully applied in many areas, such as face detection, hand-written digit recognition, and data mining, etc.

However, the SVM has two drawbacks. First, since it is originally a model for binary-class classification, we should use a combination of SVMs for multi-class classification. Methods for combining SVMs for multi-class classification have been proposed [6], but a combination of SVMs for multi-class classification does not improve the performance as much as SVM for binary classification. Second, since learning SVM is a very time-consuming for a large scale of data, we should use approximate algorithms (e.g. decomposition methods, sequential minimal optimization algorithm[8]). Using the approximate algorithms can reduce computation time, but degrade classification performance.

To overcome the above drawbacks, we propose to use the SVM ensemble. A ensemble of classifiers improve an individual classifier [5]. This implies the improvement of classification performance by using the SVM ensemble. Likewise, we also expect that the SVM ensemble will improve classification performance in case of the multi-class classification. The idea of the SVM ensemble has been proposed in [10]. They used a boosting technique to train each individual SVM and took another SVM for combining several SVMs. In this paper, we propose to use the SVM ensemble based on bagging and boosting techniques.

This paper is organized as follows. Section 2 describes the theoretical background of the SVM. Section 3 describes the SVM ensembles, the bagging and boosting method and three different aggregation methods. Section 4 shows the simulation results when the proposed SVM ensembles are applied to classification problems, such as IRIS data classification, hand-written digit recognition, and fraud detection. Finally, a conclusion is drawn.

2. Support Vector Machines

In theory, SVM classification can be traced back to the classical structural risk minimization (SRM) approach, which determines the classification decision function by minimizing the empirical risk, as

$$R = \frac{1}{l}\sum_{i=1}^{l} |f(\mathbf{x}_i) - y_i|, \tag{1}$$

where l and f represent the size of examples and the classification decision function, respectively. For SVM, the primary concern is determining an optimal separating hyperplane that gives a low generalization error. Usually, the classification decision function in the linearly separable problem is represented by

$$f_{\mathbf{w},b} = sign(\mathbf{w} \cdot \mathbf{x} + b). \tag{2}$$

In SVM, the optimal separating hyperplane is determined by giving the largest margin of separation between different classes. This optimal hyperplane bisects the shortest line between the convex hulls of the two classes. The optimal hyperplane is required to satisfy the following constrained minimization, as

$$Min : \frac{1}{2}\mathbf{w}^T\mathbf{w},$$
$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \tag{3}$$

For the linearly non-separable case, the minimization problem can be modified to allow misclassified data points. SVM can be applied to multi-class classification by combining SVMs.

3. Support Vector Machine Ensemble

An ensemble of classifiers is a collection of several classifiers whose individual decisions are combined in some way to classify the test examples [5]. It is known that an ensemble often gives a much better performance than the individual classifiers which compose it. Hansen et. al. [7] show why the ensemble gives a better performance than individual classifiers as follows. Assume that there are an ensemble of n classifiers: $\{f_1, f_2, \ldots, f_n\}$ and consider a test data $\mathbf{x}$. If all the classifiers are identical, they are wrong at the same data, where an ensemble will show the same performance as individual classifiers. However, if the classifiers are different and their errors are uncorrelated, then when $f_i(\mathbf{x})$ is wrong, most other classifiers, except for $f_i(\mathbf{x})$, may be correct. As a result, majority voting will be also correct. More precisely, if the error of individual classifier is $p < 1/2$ and the errors are independent, then the probability p_E that the result of majority voting is incorrect is $\sum_{k=\lceil n/2 \rceil}^{n} p^k(1-p)^{(n-k)}$ $(< \sum_{k=\lceil n/2 \rceil}^{n} (\frac{1}{2})^k (\frac{1}{2})^{(n-k)} = \sum_{k=\lceil n/2 \rceil}^{n} (\frac{1}{2})^n)$. When the size of classifiers n is large, the probability p_E becomes very small. Therefore, an ensemble of SVMs is expected to overcome the weakness of the performance degradation of SVM, when we use approximated learning algorithm or when we combine SVMs for multi-class classification.

3.1. Methods for Constructing the SVM Ensemble

Many methods for constructing an ensemble of classifiers have been developed. The most important consideration in constructing the SVM ensemble is that each individual SVM becomes different from another SVM as much as possible. This requirement can be met by using different training sets for different SVMs. We focus on representative methods, such as bagging and boosting.

3.1.1 Bagging

First, we will explain the bagging technique [2] to construct the SVM ensemble. In bagging, several SVMs are trained independently via a bootstrap method and then they are aggregated via an appropriate combination technique. Usually, we have a single training set $TR = \{(\mathbf{x}_i; y_i) | i = 1, 2, \ldots, l\}$. We build K replicate training data sets $\{TR_{bootstrap_k} | k = 1, 2, \ldots, K\}$ by randomly resampling, but with replacement, from the given training data set TR repeatedly. Each example $\mathbf{x}_i$ in the given training set TR may appear repeated times or not at all in any particular replicate training data set. Each replicate training set will be used to train a certain SVM.

3.1.2 Boosting

The representative boosting algorithm is the ADABOOST algorithm[11]. Like bagging, each SVM is also trained using a different training set. However, the selection scheme of training samples in the ADABOOST method is quite different from the bagging method. Figure 1 shows the pseudo code of the used ADABOOST algorithm.

3.2. Methods for Aggregating Support Vector Machines

After training, we need to aggregate several independently trained SVMs in an appropriate combination manner. We consider two types of combination techniques, such as the linear and nonlinear combination method. The linear combination method, as a linear combination of several SVMs, includes LSE-based weighting. LSE-based weighting are often used for bagging and boosting, respectively. A nonlinear method, as a majority voting and a nonlinear combination of several SVMs, includes the double-layer hierarchical combining that use another upper-layer SVM to combine several lower-layer SVMs.

3.2.1 Majority Voting

Majority voting is the simplest method for combining several SVMs. Let $f_k (k = 1, 2, \ldots, K)$ be a decision function of the kth SVM in the SVM ensemble and $C_j (j =$

```
Input:
    A set $TR$ of $l$ labeled examples:
        $S = \{(\mathbf{x}_i; y_i), i = 1, 2, \ldots, l\}$,
    Labels: $y_i \in Y = \{1, \ldots, C\}$.
$p_0(\mathbf{x}_i) := 1/l$.
for $k = 1$ to $K$
    Build $TR_{boost_k} = \{(\mathbf{x}_i; y_i) | i = 1, 2, \ldots, l\}$
        based on the $p_{k-1}(\mathbf{x}_i)$.
    Train the $k$th SVM $h_k$ using $TR_{boost_k}$.
    $\epsilon_k := \sum_{i=1}^{l} p_k(i) |\{i | h_k(\mathbf{x}_i) \neq y_i\}|$.
    $\alpha_k := \frac{1}{2} ln(\frac{\epsilon_k}{1-\epsilon_k})$.
    for $i = 1$ to $l$
        $p_{k+1}(\mathbf{x}_i) = \frac{p_k(\mathbf{x}_i)}{Z_k} \times \begin{cases} \exp(-\alpha_k) & \text{if } h_k(\mathbf{x}_i) = y_i, \\ \exp(\alpha_k) & \text{if } h_k(\mathbf{x}_i) \neq y_i. \end{cases}$
        where $Z_k$ is a normalization factor
            to make $\sum_{i=1}^{l} p_{k+1}(\mathbf{x}_i) = 1$.
    end
end
```

Figure 1. The AdaBoost algorithm.

$1, 2, \ldots, C)$ denote a label of the j-th class. Then, let $N_j = \#\{k | f_k(\mathbf{x}) = C_j\}$, i.e. the number of SVMs whose decisions are known to the jth class. Then, the final decision of the SVM ensemble $f_{mv}(\mathbf{x})$ for a given test vector $\mathbf{x}$ due to majority voting is determined by

$$f_{mv}(\mathbf{x}) = \underset{j}{argmax} N_j. \tag{4}$$

3.2.2 LSE-based Weighting

LSE-based weighting treats several SVMs in the SVM ensemble with different weights. Often, the weights of several SVMs are determined in proportion to their accuracy of classification. Here, we propose to learn the weights using the LSE method as follows.

Let $f_k(k = 1, 2, \ldots, K)$ be a decision function of the kth SVM in the SVM ensemble that is trained by a replicate training data set $Thau_k^B = \{(\mathbf{x}_i'; y_i') | i = 1, 2, \ldots, L\}$. The weight vector $\mathbf{w}$ can be obtained by $\mathbf{w}_E = \mathbf{A}^{-1}\mathbf{y}$, where $\mathbf{A} = (f_i(\mathbf{x}_j))_{K \times L}$, and $\mathbf{y} = (y_j)_{1 \times L}$. Then, the final decision of the SVM ensemble $f_{mv}(\mathbf{x})$ for a given test vector $\mathbf{x}$ due to the LSE-based weighting is determined by

$$f_{LSE}(\mathbf{x}) = sign(\mathbf{w}_E \cdot [(f_i(\mathbf{x}))_{K \times 1}]). \tag{5}$$

3.2.3 Double-layer Hierarchical Combining

We can use another SVM to aggregate the outputs of several SVMs in the SVM ensemble. This combination consists of a double-layer of SVMs hierarchically, where the outputs of several SVMs in the lower layer feed into a super SVM in the upper layer. Let $f_k(k = 1, 2, \ldots, K)$ be a decision function of the kth SVM in the SVM ensemble and F be a decision function of the super SVM in the upper layer. Then, the final decision of the SVM ensemble $f_{SVM}(\mathbf{x})$ for a given test vector $\mathbf{x}$ due to the double-layer hierarchical combining is determined by

$$f_{SVM}(\mathbf{x}) = F((f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_K(\mathbf{x}))), \tag{6}$$

where K is the number of SVMs in the SVM ensemble.

4. Simulation Results and Discussion

4.1. UCI Hand-written Digit Recognition

There is some literature on handwritten digit recognition using SVM [4]. We used the UCI hand-written digit data, containing a traingin set of 3,828 and a test set of 1,797 [1]. The original image of each digit having the size of 32×32 pixels was reduced to the size of 8×8 pixels. For both bagging and boosting, we used 1000 samples, and an ensemble of 10 multi-class SVMs. Each multi-class SVM used the one-against-one multi-classification method, and so had 45 SVMs. A 2-d polynomial kernel was used in each SVM. Table 1 shows the classification results. To circumvent the tweak problem, a test for each classifier was performed in 10 independent runs of simulation and the average performance is reported in the table.

Table 1. The correct classification rates of UCI hand-written digit recognition.

	Normal	
Single SVM	96.02%	
	Bagging	Boosting
SVM E. (Majority voting)	96.85%	97.15%
SVM E. (LSE-based weighting)	97.27%	97.61%
SVM E. (Hierarchical combining)	97.38%	97.83%

4.2. Fraud Detection

We often witness many occurrences of customer fraud in our society [9]. Here, we tackled mobile telecommunication payment fraud detection using the proposed SVM ensemble. We used the database obtained from a mobile telecom company. It recorded one year's action data of 53,696 customers. we extracted 8 salient features to measure customer payment behaviors. The data set was divded into a training set containing 70% of it and a test set 30% of it. The fraud detection problem can be a binary or a multi-class classification problem. If customers are to be divided into two classes: fraud or non-fraud, fraud detection is a binary classification problem. If customers are to be classified into more than two (four, here) confidence grades, fraud detection is a multi-class classification problem. We performed simulations for both kinds of fraud detection. For both bagging and boosting, we used samples of 30% of the training set, and an ensemble of 11 SVMs for the binary-class case or 11 muiti-class SVMs for the multi-class case. Each multi-class SVM for the multi-class case used the one-against-one multi-classification method, and so has 6 SVMs. A 3-d polynomial kernel or RBF kernel was used for each case. We used only a 3-d polynomial kernel for the hierarchical combining method. LSE-based weighting was not applied beacause of high computational complexity.

Table 2 and Table 3 show the classification results in the case of binary-class and multi-class fraud detection, respectively. For boosting for a single SVM, we used the last SVM obtained in boosting. circumvent the tweak problem, a test of each classifier was performed in 10 independent runs of simulation and the average performance is reported in the tables.

Table 2. The correct classification rates of the binary-class fraud detection.

	Normal	Boosting
Single SVM (Poly.)	84.95%	89.91%
Single SVM (RBF)	83.97%	89.83%
	Bagging	Boosting
Majority Voting (RBF)	93.49%	96.61%
Majority Voting (Poly.)	95.75%	97.28%
Hierarchical SVM (Poly.)	84.38%	86.97%

Table 3. The correct classification rates of the multi-class fraud detection.

	Normal	Boosting
Single SVM (Poly.)	76.53%	87.18%
Single SVM (RBF)	79.78%	88.68%
	Bagging	Boosting
Majority Voting (RBF)	88.89%	89.65%
Majority Voting (Poly.)	93.52%	96.43%
Hierarchical SVM (Poly.)	81.15%	82.08%

5. Conclusion

To overcome the weakness of the performance degradation when we use approximated learning algorithm or when we combine SVMs for multi-class classification, we addressed the SVM ensemble that consists of several independently trained SVMs. We evaluated the classification performance of the proposed SVM ensemble over hand-written digit recognition and fraud detection. The SVM ensembles outperform a single SVM for all applications in terms of classification accuracy. For ensemble construction methods, the boosting methods provides a better classification performance than the bagging method. For three different aggregation methods, double-layer hierarchical is the best for handwritten digit recognition, and majority voting is the best for fraud detection. Therefore, the SVM ensemble is expected to improve a single SVM for many real applications, such as large data problems, high-dimensional data problems, and multi-class classification problems.

Acknowledgements

The authors would like to thank the Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program.

References

[1] B. D. Bay. The uci kdd archive. *[http://kdd.ics.uci.edu]. Irvine, CA: University of Califonia*, 1999.
[2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
[3] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
[4] V. V. C. Cortes. Support vector network. *Machine Learning*, 20:273–297, 1995.
[5] T. G. Dietterich. Machine learning research: Four current directions. *The AI Magazine*, 18(4):97–136, 1998.
[6] C. W. J. Weston. Support vector machines for multi-class pattern recognition. *Proceedings of the 7th European Symposium on Artificial Neural Networks*, 1999.
[7] P. S. L. Hansen. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993–1001, 1990.
[8] J. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods: Support Vector Machines*, 1999.
[9] F. P. Tom Fawcett. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1, 1997.
[10] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1999.
[11] R. S. Y. Freund. Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.