A Consensual Subspace Method to Enhance Classification Accuracy

Tzu-Cheng Chuang
School of ECE
Purdue University
West Lafayette, IN, U.S.A.

Okan K. Ersoy
School of ECE
Purdue University
West Lafayette, IN, U.S.A.

Saul B. Gelfand
School of ECE
Purdue University
West Lafayette, IN, U.S.A.

Abstract

The proposed method uses different input features to partition the sample space in to subspaces in a two-level decision treelike structure to enhance the performance of a classifier. The support vector machine is used as the classifier in this paper. Each input feature used is associated with a threshold such that an input vector traverses to either the right node or the left node of a parent node. Given a feature, the best threshold is usually found by minimizing a measure such as the impurity, characterized by Gini index or information entropy. In this way, for each pair of a feature and the corresponding threshold, the data is partitioned in to two groups. Each group is trained with a specialized SVM. During testing, each data point is directed to one of the SVM's based on the feature used and its threshold. The method is further generalized by choosing a subset of rank-ordered features. For this purpose, an impurity measure is used. In this way, a number of subspace classifiers are generated. In the end, the final classification is done by consensus between the subspace classifiers. This usually results in better accuracy as compared to a single SVM classification.

Introduction

The method proposed in this paper has certain features which are common to decision trees first developed in the 1980s. Two well-known decision trees are CART (Breiman, 1984) and C4.5 (Quinlan, 1992). At each node in a decision tree, the algorithm searches for the best splitting point among features which has the maximum reduction of the impurity measure, such as Gini criterion or information gain. Then, the splitting process is repeated until the number of samples at a node is less than a specified number or the data samples in that node belong to the same class.

In C4.5, the best splitting point is picked by a greedy search to find the maximum reduction of impurity. That method only considers splitting along a Cartesian axis. CART with linear combination (CART-LC) and OC1 (Murthy, 1994) use an oblique hyperplane. It is especially hard to find the best hyperplane in a high dimensional dataset. They use the method called perturbation of the coefficient to find the best hyperplane. After a number of trials, if the impurity is not improved anymore then the hyperplane coefficients are stored. These coefficients may represent just a local optimum, and not the global optimum. Yildiz (Yildiz and Alpaydin, 2000) proposed a method to combine the oblique hyperplane and the axis-parallel hyperplane. If the oblique hyperplane gives better impurity measure at a node, then their method uses the oblique hyperplane. They found that only 10% of the nodes used an oblique hyperplane. Since the axis-parallel method is easier to implement, and there is no big difference between these two ways, the axis-parallel method is used in the proposed method in this paper.

The support vector machine (SVM) (Vapnik, 1995) is a well-known and widely-used technique for machine learning. It involves optimization to find the best separation hyperplane between two classes. One method which combines SVM's and

decision trees is called LSVM-DT (Chi and Ersoy, 2002). In that paper, the authors use SVM to find the best hyperplane in each node. In the rare event case, which means one class is more dominant than the other one, it randomly chooses data samples with replacement from the rare class and adds them to the training set.

It was shown by Ho (1998) that we can randomly choose part of the feature space as new input vectors to train a number of classifiers. After generating a number of different classifiers, consensus between them usually improves classification. This is an example of bagging methods. The proposed method also targets generation of a number of classifiers followed by consensus. For this purpose, dominant features defined in terms of an impurity measure are used to split the sample space in to two subspaces.

Support Vector Machine

Vapnik invented SVM's with a kernel function in the 1990s (Vapnik, 1995). This algorithm is initially designed for the two-class classification problem. One class output is marked as 1, and the other class output is marked as -1.

The algorithm tries to find the best separating hyperplane with the largest margin width. By generating a wide margin hyperplane from the training samples, it is expected to achieve better testing accuracy. In the SVM, the hyperplane of a non-separable classification problem is determined by solving the following equation:

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \\ &\textit{subject to} \\ &y_i(x_i^T w + b) \geq 1 - \xi_i \\ &\xi_i \geq 0 \end{aligned} \tag{1}$$

where x_i is the i^{th} data vector, y_i is the binary (-1 or 1) class label of the i^{th} data vector, ξi is the slack variable, w is the weight vector normal to the hyperplane, C is the regularization parameter, and b is the bias. It can be shown that the margin width is equal to $2/|w|$.

Usually the original data is mapped by using a kernel function to a higher dimensional representation before classification. Some common kernel functions are linear, polynomial, radial basis and sigmoid functions.

In the experiments conducted, the SVM-Light (Joachims, 2004) software was utilized. We used linear kernel function and picked C equal to 1 in these experiments.

Impurity Measure

In order to partition the data into 2 groups, the impurity measure is to be used to decide which splitting point is best. The reduction of impurity is given by

$$\Delta I = I_{initial} - P(g_1)I(g_1) - P(g_2)I(g_2) \tag{2}$$

where $I_{initial}$ is the impurity without splitting, $P(g_1)$ is the probability that data samples fall in group 1 and $I(g_1)$ is the impurity measure for group 1.

Two possible impurity measures are defined next. The Gini criterion is defined as follows:

$$I_{gini} = \sum_{i \neq j} p(i)p(j) = 1 - \sum_i p(i)p(i) \tag{3}$$

where $p(i)$ is the probability of class *i*.

Information gain or entropy is defined as follows:

$$I_{entropy} = -\sum_{i} p(i)\log_2(p(i)) \tag{4}$$

Consensual Subspace Method

As in a decision tree, the best splitting point for each feature is first searched for. This is done by greedy search. For a splitting point, the reduction of impurity is computed. Then, the best threshold related to each feature resulting in most reduction of impurity is determined. Based on the current feature and its related threshold, the data samples are thus partitioned in to two subspaces. In each subspace, a classifier (SVM in this paper) is used to classify the training samples falling in that subspace. After training, each SVM finds the best separation hyperplane in the related subspace. It is actually possible to compute the reduction of impurity again in each such subspace. This would be related to the expected testing accuracy. This process can actually be recursively continued to build a complete decision tree. However, in this paper, we only consider the first and the second stages. In the first stage, the subspaces are generated. In the second stage, the classifiers (SVM's) do the classification in each subspace. This is followed by consensus between the results obtained with different rank-ordered features.

The algorithm is further given in detail below with a 2-class example.

Training Algorithm

Input: 2-class data

Variables: $(f_j, t_j, I_j), j = 1,...,m$

For each feature $f_j, j = 1,...,m$

Find the best splitting point t_i for this feature so that it reduces the impurity most.

$$\Delta I = I_{initial} - P(g_1)I(g_1) - P(g_2)I(g_2)$$

Store the best splitting point t_j for each f_j

For each feature $(f_j, t_j), j = 1,...,m$

Train 2 SVM's, one is by $(\vec{x_i}, y_i)$ s.t. $x_{ij} < t_j$, and the other is by $(\vec{x_i}, y_i)$ s.t. $x_{ij} \geq t_j$.

For each SVM, the classifier separates the data samples into 2 regions again.

$$I_i = I_{initial} - P(g_{1,1})I(g_{1,1}) - P(g_{1,2})I(g_{1,2}) - P(g_{2,1})I(g_{2,1}) - P(g_{2,2})I(g_{2,2})$$

Store the best splitting point t_j for each f_j and the maximal drop of impurity I_j

Sort (f_j, t_j, I_j) with descending order of I_j. This gives the ranking of the input features.

The process of segmentation in to two subspaces and subsequent classification is depicted in Figure 1. In practice, only a number of the most important features are used. The procedure for subsequent consensus between the results of the classifiers is further discussed below.

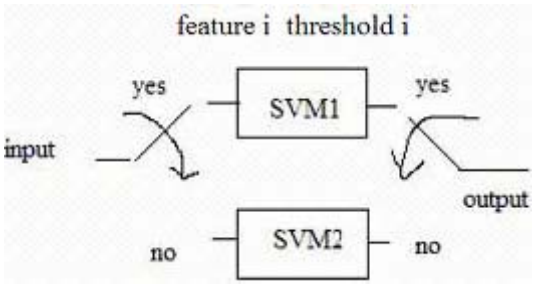

Figure 1. The process of segmentation into two subspaces and subsequent classification.

During testing, each sample is classified by one SVM, either SVM1 or SVM2 based on a feature *i* and its threshold. There are a number of methods to do consensus between the results of the different classifiers. One method is the majority voting rule to aggregate the results of those classifiers, and, in a two-class problem, it is given by

$$C^*(\vec{x}) = sign(\sum_i C_i(\vec{x})) \tag{5}$$

where $C_i(\vec{x})$ is the classification for the sample $\vec{x}$ from classifier *i*.

Another method is to weigh each classifier result by least-squares weighting as shown in Eq. (6). The weights are found using the training set by the least-squares method.

The training set is $\{(\vec{x_i}, y_i) \mid i = 1,...,L\}$

There are *K* classifiers.

$$A = \left[f_1(\vec{x}) \quad ... \quad f_K(\vec{x}) \right]_{L \times K}, \ \vec{y} = (y_i)_{L \times 1} \text{ (desired outputs)}$$
$$\vec{w} = A^{-1}\vec{y} = [w_1,...w_k]^T \tag{6}$$
$$C^*(\vec{x}) = sign(\sum_i w_i C_i(\vec{x}))$$

Experimental Results

We used a synthetic dataset and some real datasets to test our algorithm. The summary of the datasets is shown in Table 1.

Table 1. Summary of the datasets.

Dataset Name	# of samples	# of features	# of classes
Bc_wisconsin	683	9	2
Breast cancer	286	9	2
Bupa liver disorders	345	6	2
Ringnorm	7400	20	2

The datasets are downloaded from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) except for the synthetic ringnorm dataset. The ringnorm dataset is downloaded from Delve's website (http://www.cs.toronto.edu/~delve/data/). The Ringnorm dataset is first used by Leo Breiman (Breiman 1996a). It is a 20-dimensional, 2 class classification problem. Each class is drawn from a multivariate normal distribution. Class 1 has mean zero and covariance equal to 4 times the identity matrix. Class 2 has mean (a,a,..a) and unit covariance with a = 2/sqrt(20).

We experimented with the synthetic ringnorm dataset first. The results showed that the ranking of the features and using only some of them is important to increase testing accuracy. How many features to use is data dependent.

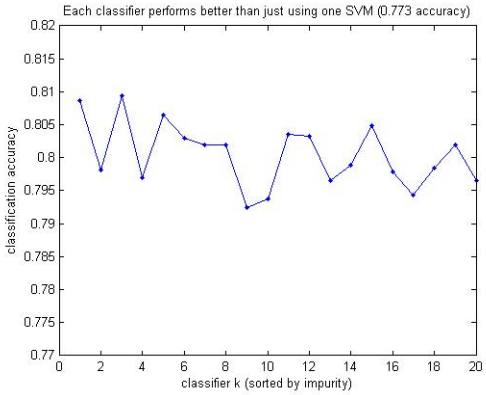

Figure 2. The subspace SVM classifier k is sorted by the impurity measure.

With the ringnorm dataset, the classification accuracy of a single SVM was 0.773. From Figure 2, we see that all of the subspace classifiers perform better than a single SVM. All of the classification accuracies obtained from subspace SVM's are higher than 0.773. The higher ranked classifiers with better reduction of impurity also typically have higher classification accuracy. The consensus results by least-squares weighting are shown in Figure 3, and the consensus results by the majority voting rule are shown in Figure 4.

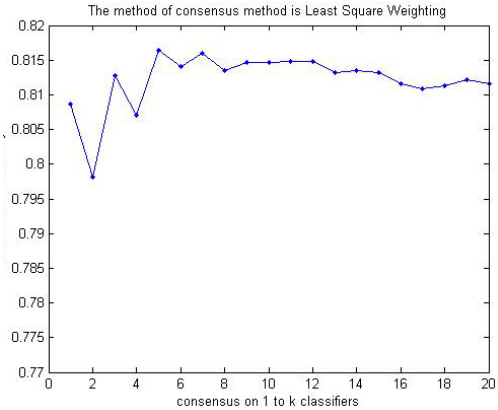

Figure 3. The consensus is done by least-squares weighting among k classifiers.

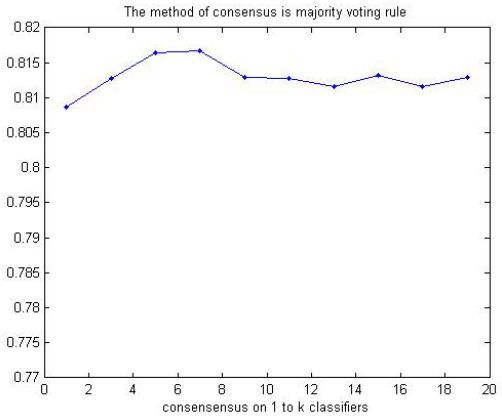

Figure 4. The consensus is done by majority voting rule among k classifiers.

From Figure 3 and Figure 4, it is observed that it is better to do the consensus with a limited number of input features. We can see that the majority voting rule (MVR) gives us smoother curve on the consensus. By using this smoother curve, the worse consensus result can be avoided. Therefore, we used MVR in our subsequent experiments on other datasets. In order to avoid the tie situation, we chose the number k for MVR consensus an odd number. We picked the best ¼ of the subspace SVM's for consensus to get the final decision.

Table 2. 5x2 Cross Validation results.

Classification accuracy	Subspace SVM		Bagging		Partition_bagging		single SVM	
	Avg(%)	Std(%)	Avg(%)	Std(%)	Avg(%)	Std(%)	Avg(%)	Std(%)
bc_wisconsin	96.34	0.66	96.72	0.67	96.49	0.48	96.54	0.67
breast_cancer	72.38	3.43	70.63	2.99	70.77	2.23	71.40	2.67
bupa	68.23	2.14	68.35	2.87	68.23	2.73	67.25	2.61
ringnorm	81.73	0.42	77.07	0.57	77.06	0.55	77.06	0.60

In Table 2, we compared the proposed algorithm with bagging (Breiman 1996b), partition bagging (Dong and Han, 2005) and single SVM by using 5x2 cross validation (Alpaydin, 1999). Bagging is done by sampling the training data uniformly and with replacement. Partition bagging is done by randomly partitioning the training data into several partitions, and using the smaller datasets to train the classifier. In this experiment, the size of the partition was chosen equal to 2. The number of classifiers for bagging and partition bagging is 2 times the number of the features plus 1. The final aggregation was done by the majority voting rule.

From Table 2, we observe that the proposed subspace method has better performance than a single SVM. Compared with the bagging approaches, it is also competitive with them. It even performed better than bagging in 2 datasets.

Discussions

The example shown in Figure 2 is a special case that all subspace SVM's perform better than a single SVM. We've observed that sometimes some of the subspace

SVM's might not perform well. By using the best ¼ of the subspace SVM's and consensus among them can recover and provide good results.

Bagging is a simple idea to generate different independent training sets. By changing the distribution of the input spaces, several classifiers are obtained. Our method also changes the input space. We do not just reshuffle the training set randomly, but we use a systematic way to find the best splitting point along one feature. Based on each feature and splitting point, the training data and testing data use this criterion to decide which SVM it should adopt. These different splitting spaces for each subspace SVM make each of them generate independent classification.

Subspace SVM requires the extra effort to find the splitting point and generate different SVM's for later consensus. The number of subspace SVM's is at most equal to the number of features. This is very unlike bagging, that the number of new training sets is picked or tuned by someone. The higher classification accuracy is obtained after extra computation is used.

For the bc_wisconsin dataset, originally single SVM gives 96% classification accuracy. Since the classification is good enough, it's really hard to make big improvement on it.

Conclusions

The proposed method uses input features to segment the sample space in to a number of regions based on an impurity measure. Different partitions based on different features result in different classifiers. For this purpose, it is advisable to use only the most important input features. Subsequent consensus between the classifier outputs yield an overall classifier which usually performs better than a single classifier without sample space segmentation. The method is competitive with other sampling methods such as bagging.

This paper focused on the binary classification problem. The method can be easily generalized to the multi-class problems, for example, based on one-against-all or one-against-one binary classifiers. Among those predictions by different classifiers, the consensus can be done by using the highest rank method, the Borda count method or logistic regression. (Ho 1994)

Acknowledgement

This research was supported by NSF Grant MCB-9873139 and partly by NSF Grant #0325544.

References

Alpaydin, E., 1999," Combined 5 × 2 cv F Test for Comparing Supervised Classification Learning Algorithms," *Neural Computation.* Vol. 11, pp.1885-1892.

Breiman, L., 1984, "Classification and regression trees ," Chapman & Hall.

Breiman, L., 1996a, "Bias, variance and arcing classifiers," Tec. Report 460, Statistics Department. University of California.

Breiman, L., 1996b, "Bagging Predictors," Machine Learning, Vol. 24, No. 2, pp. 123-140.

Chi, H. M. and Ersoy, O. K., 2002, "Support Vector Machine Decision Trees with Rare Event Detection," *International Journal of Smart Engineering System Design*, Volume 4, Issue 4, pp. 225 – 242.

Dong, Y.S. and Han K.S., 2005, "Boosting SVM Classifiers By Ensemble," Posters of the 14th international conference on World Wide Web, Chiba, Japan , pp. 1072 – 1073.

Ho, T.K., Hull, J.J., Srihari, S.N., "Decision Combination in Multiple Classifier Systems," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 1, pp. 66-75, Jan., 1994.

Ho, T. K., 1998, "The Random Subspace Method for Constructing Decision Forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832-844.

Joachims, Thorsten, 2004, http://www.cs.cornell.edu/People/tj/svm_light/ .

Murthy, S. K., S. Kasif, and S. Salzberg, 1994. "A system for induction of oblique decision trees," *Journal of Artificial Intelligence Research 2*, pp.1-33.

Quinlan, J. R., 1993, "C4.5: Programs for Machine Learning." Morgan Kaufmann Publishers.

Vapnik, V., 1995, "The Nature of Statistical Learning Theory." Springer-Verlag,

Yildiz, O. T. and Alpaydin E., 2000, "Linear discriminant trees." Proceedings of 17th International Conference on Machine Learning, pp.1175- 1182.