An Evidence Theoretic Ensemble Design Technique

H. Altınçay
Department of Computer Engineering, Eastern Mediterranean University
KKTC, Mersin 10, Turkey
E-mail:hakan.altincay@emu.edu.tr

Abstract

Ensemble design techniques based on resampling the training set are successfully used to improve the classification accuracies of the base classifiers. In Boosting technique, each training set is obtained by drawing samples with replacement from the available training set according to a weighted distribution which is iteratively updated for generating new classifiers for the ensemble. The resultant classifiers are accurate in different parts of the input space mainly specified the sample weights. In this study, a dynamic integration of boosting based ensembles is proposed so as to take into account the heterogeneity of the input sets. In this approach, a Dempster-Shafer theory based framework is developed to consider the training sample distribution in the restricted input space of each test sample. The effectiveness of the proposed technique is compared to AdaBoost algorithm using nearest mean type base classifier.

1 Introduction

Boosting is a popular ensemble creation technique which takes into account the classification results of the previous classifiers to construct additional ones. The sequential structure of the algorithm allows to create new classifiers which are more effective on the training samples that the current ensemble has a poor performance. In order to achieve this, weighting is applied on the training samples where a training sample with a high weight has a larger probability of being used in the training set of the next classifier. The weights are updated in an iterative manner so that new classifiers mainly focus on the samples difficult to classify. AdaBoost is the most popular boosting algorithm.

In AdaBoost technique, the reliability of the classifier outputs is dependent on the input due to the weighted resampling [1]. For instance, the decision of a classifier for an input lying in a restricted space that is resampled by a large number of times is expected to be reliable. However, this may not be true for the input spaces which have no representatives in the resampled training set. Hence, the fact that the classifiers developed using AdaBoost may be accurate in some regions of the input space should be considered during the combination operation. The weighted majority voting rule used in the AdaBoost algorithm does not take into account this input dependent information. Dynamic combination schemes that take into account the distribution of the training samples in the restricted space where the given test sample lies may provide better results.

In order to take into account the distribution of the training samples in different parts input space, a Dempster-Shafer theory based (evidential) pattern classification technique is proposed by Denæux [2]. In that approach, each neighbor of a given test sample in the training set is considered as a piece of evidence supporting the class that the training sample belongs. The basic probability assignments from all neighbors are then combined to predict the class of the tested sample. In boosting technique, weights for different samples are naturally available where the ultimate aim is analogous; each training sample has a different influence on the decision depending on the difficulty of its classification since, more replicas of difficult samples are used than easier ones in classifier training. However, the sample weights are explicitly considered only in the training phase. In this study, an evidence-theoretic framework for boosting is proposed so as to take into account the weights and distances of the neighboring training samples in both training and testing boosting based ensembles. In the proposed approach, the weight update mechanism of AdaBoost is preserved where a weight and distance dependent belief structure assignment is developed. The proposed approach is used for boosting nearest mean classifier (NMC) where better accuracies than AdaBoost are obtained.

2 Evidential Pattern Classification

Let Ω denote the set of class labels and $\mathcal{S} = \{(x_n, y_n)\}$, $n = 1, \ldots, N$ be the set of training samples where x_n denotes the nth input sample and $y_n \in \Omega$ is its label. Given a test sample x, each training sample is considered to provide a piece of evidence about the class label of x. In other words, each $x_n \in \mathcal{S}$ induces a belief structure m_n with two focal elements, $\{w_q\}$ and

Ω where $w_q \in \Omega$ is the class that the training sample x_n belongs as,

$$m_n(\{w_q\}) = \alpha\phi_q(||x_n - x||) = \alpha e^{-\gamma_q||x_n - x||^2}$$
$$m_n(\Omega) = 1 - m_n(\{w_q\}) \quad (1)$$

where $||x_n - x||$ is the Euclidean distance between x_n and x. The class-independent design parameter α and the class-dependent parameter γ_q determine the way basic probability values are assigned to $\{w_q\}$ and Ω. As α increases, the evidence provided is considered to be more certain. The influence of the distance on basic probability assignments is class dependent and it is adjusted by γ_q. As the distance increases, more probability mass is assigned to Ω. These belief structures are combined using Dempster's rule of combination [3]. In order to make a joint decision, pignistic probabilities can be computed as $P_{Bet}(w_i) = \sum_{w_i \subseteq B} \frac{m(B)}{|B|}$, $\forall w_i \in \Omega$, where B denotes the focal elements getting nonzero basic probability value [4]. Then, the class getting the maximal $P_{Bet}(.)$ value can be selected as the joint decision.

In summary, the main idea is to treat each neighboring sample as a piece of evidence for the class label of the tested sample. In this paper, this idea is used to propose an evidence-theoretic framework for boosting where the sample weights are explicitly used to compute the measurement level classifier outputs.

3 Evidence Theoretic Framework for Boosting

Let $W_t(n)$ denote the weight of the nth training sample in $\mathcal{S}$ initialized to $1/N$ and t denotes the iteration count. $\Omega = \{w_a, w_b\}$ since a 2-class problem is considered. Let $\mathcal{S}_t$ denote bootstrap sample set obtained by drawing with replacement N samples from $\mathcal{S}$ using distribution W_t. $d_n(j)$ is defined as the Euclidean distance of x_n to its jth nearest neighbor in $\mathcal{S}$ denoted by $neig_n(j)$. $d_n(1)$ is zero which is the distance of the sample with itself and $d_n(2)$ is the distance from the closest different sample.

The proposed algorithm named as E-Boost is given below. The initial classifier makes use of equal sample weights. In each iteration, the weights of the correctly classified training samples are decreased. Given the current weight vector W_t, the training set is resampled to generate a new ensemble member. Then, each training sample x_n is classified by taking into account its k-nearest neighbors. For this purpose, each neighboring sample having the *same* class label as the classifier decision (totally, $k' \leqslant k$) induces a belief structure m_j with focal elements $\{w_q\}$ being the decision of the trained classifier and Ω. The main idea behind this is to compute the total support on the output of the classifier. It should also be noted that k' is different for each training sample.

for $t = 1, ..., T$

Build classifier $\mathcal{C}_t$ using sample set $\mathcal{S}_t$ resampled from $\mathcal{S}$ using distribution W_t.

for n = 1 to N

Compute the most likely class, $w_q = C_t(x_n)$

Compute the k-nearest neighbors of x_n in $\mathcal{S}$ using $d_n(j)$, $j = 1, \ldots, k$.

for j = 1 to k' *// consider neighbors of the same class as* w_q

$m_j(\{w_q\}|neig_n(j)) = f(q, W_t(neig_n(j)), d_n(j))$

$m_j(\Omega|neig_n(j)) = 1 - m_j(\{w_q\}|neig_n(j))$

end

$m^n_{comb}(.) = m_1(.|neig_n(1)) \oplus \ldots \oplus m_{k'}(.|neig_n(k'))$

Compute pignistic probability as,

$$P_{Bet}(w_q) = m^n_{comb}(w_q) + m^n_{comb}(\Omega)/2$$

The combined decision is computed as,

$$C_t(x_n) = \begin{cases} w_q & if\ |P_{Bet}(w_q) - P_{Bet}(w'_q)| > \tau \\ rand(\{w_a, w_b\}) & otherwise \end{cases} \quad (2)$$

Calculate the weighted error using $\epsilon_t = \frac{1}{N} \sum_{n=1}^{N} W_t(n)(1 - q_{n,t})$ where $q_{n,t} = 1$ if x_n is correctly classified and zero otherwise.

Compute $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$, $\epsilon_t \in (0, 0.5)$ and update the weights using

$W_{t+1}(n) = \frac{W_t(n)}{Z_t} e^{-\alpha_t}$ if $C_t(x_n) = y_n$, where Z_t is a normalization factor so that W_{t+1} is a distribution.

end

end

The basic probability assignment is defined as,

$$m_j(\{w_q\}|neig_n(j)) = \frac{(W_t(neig_n(j))/W_{t,max})}{1 + e^{\frac{1}{d^q_{avg}}(d_n(j) - d^q_{avg})}} \quad (3)$$

where $W_{t,max}$ is the maximum weight value at the tth iteration. d^q_{avg} used in the denominator is the within-class average of k-nearest neighbor Euclidean distances for the class w_q. As seen in the equation, the basic probability value is proportional to the weight of the neighbor; higher weights correspond to stronger evidence. The weights are normalized by their maximum values so that $m_j(\{w_q\}|neig_n(j)) < 1$. The denominator also depends on the decided class. As the distance increases above the average d^q_{avg}, the evidence provided is considered to decrease whereas a smaller distance corresponds to an increasing evidence.

k' belief structures are then combined using Dempster's rule of combination to compute the combined belief structure, $m^n_{comb}(.)$. The pignistic probabilities obtained from the combined belief structure are used to select the most likely pattern class. If the decision of the classifier is w_a, it is expected that $P_{Bet}(w_a) >> P_{Bet}(w_b)$ where the support on w_b comes from the basic probability assigned to Ω. In the proposed algorithm, the threshold τ is used to make sure that there is enough support on the decision of the classifier. If $P_{Bet}(w_a) \approx P_{Bet}(w_b)$, the classifier is not considered to provide reliable information about the class of the

sample under concern. Also, it may be the case that $P_{Bet}(w_a) = P_{Bet}(w_b) = 0$ due to no local support. In both of these cases, the decision is randomly selected.

The decisions generated by the tth classifier after combining the evidence from each k'-nearest neighbors are used to compute the weighted error, ϵ_t as in AdaBoost. If $\epsilon_t \in (0, 0.5)$, the weights of the correctly classified samples are updated. In the proposed approach, the weights of misclassified samples are not increased since, it is observed that this may easily lead to classifiers with higher error rates than 0.5. Such an update was also considered for AdaBoost and named as conservative AdaBoost. When the condition that $\epsilon_t \in (0, 0.5)$ is not satisfied, the algorithm is terminated. The output of the algorithm is a set of T different weight vectors, T classifiers and the α_t values obtained for each weight vector.

During testing an unseen input vector, combined belief structures $m_{comb}(.)$ are computed for each different member classifier. The input vector is firstly tested by each classifier and then each k' nearest neighbors having the same class label as the most likely class generated by the classifier induces a belief structure. k' belief structures are then combined using Dempster's rule of combination to compute the combined belief structure, $m_{comb}(.)$. The pignistic probabilities obtained from the combined belief structures, $\{P^t_{Bet}\}_{t=1}^T$ are aggregated to compute the resultant pignistic probabilities using weighted averaging as,

$$P^{avg}_{Bet}(w_i) = \frac{1}{T}\sum_{t=1}^{T} \alpha_t P^t_{Bet}(w_i) \qquad (4)$$

Then, the class assigned to the tested pattern is the one getting the highest $P^{avg}_{Bet}(.)$ value. Since different belief structures are obtained using resampling from the same training set, Dempster's rule of combination cannot be used. Due to the commutativity property of averaging and the linear relationship between credal level information (using basic probability values) and pignistic probabilities, averaging is considered to be a good candidate for bagging evidential k-NN classifiers [5]. Following this reasoning, the weighted form of averaging as given above is used in this study.

4 Experiments

In order to investigate the benefits of the proposed evidential framework for boosting, experiments are conducted on artificial and real data sets from UCI machine learning repository and ELENA database.

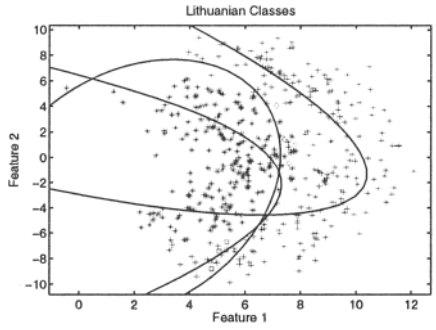

Fig. 1. The scatter plot of training samples for 2-D Lithuanin classes.

4.1 Experiments on Artificial Data Sets

Two different artificial data sets are generated using the PRTOOLS toolbox [6]. In both of these experiments, we set $k = 5$ and $\tau = 0$. For the first 2-D classification problem, Lithuanin classes are generated. For each class, 200 samples are used for training and 200 for testing. AdaBoost is run to generate an ensemble of three normal densities based quadratic classifiers. The developed classifiers are illustrated on the scatter plot of the training data in Figure 1. The first class is represented using +'s and the second class samples are represented using *'s. The test samples that are correctly classified by the proposed algorithm but misclassified by AdaBoost are also marked on the figure. □'s represent such test patterns from the first class and ◇'s represent those belonging to the second class. As seen in the figure, the proposed technique is more effective on the difficult samples lying on the border of the classes. Three of the differently classified test samples belonging to the first class and five belonging to the second class are misclassified by two of the three classifiers. This is the main reason for AdaBoost to be unsuccessful for these samples. However, they are correctly classified by E-boost due to the support from their neighbors. In this experiment the classification accuracy of the base classifier is 91.25%, where 93.50% and 95.75% accuracies are achieved by AdaBoost and proposed algorithm respectively.

For the second 2-D classification problem, 'banana' classes are generated where the same number of training and test samples are considered. Figure 2 illustrates ten NMC type ensemble members obtained using AdaBoost. Since the NMC type classifier is a weak one, it is less sensitive to changes in the training set providing decision boundaries that may be close to some others leading to correlated decisions and only 1% improvement. However, making use of the different weights of the training

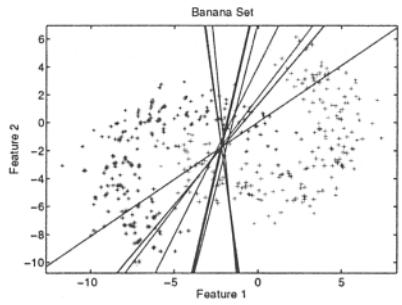

Fig. 2. Ten NMC type ensemble members obtained for 'banana' data set using AdaBoost algorithm.

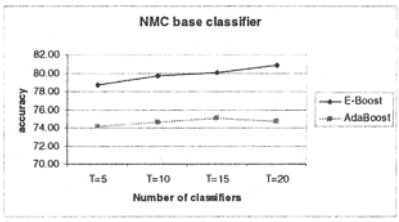

Fig. 3. Average accuracies achieved by E-boost and AdaBoost for $T = 5$, 10, 15 and 20 over eleven data sets.

samples explicitly during testing reduces the correlations providing 14.75% improvement over the base classifier.

4.2 Experiments on Real Data Sets

In order to evaluate the proposed algorithm, experiments are conducted on nine different data sets in the UCI machine learning repository and two in the ELENA data set. The experiments are repeated ten times and the average accuracies are computed. In order to evaluate the proposed framework on real data sets, nearest mean classifier (NMC) is selected as the base classifier. The experimental results for $T = 10$, $k = 5$ and $\tau = 0$ are presented in Table 1. As seen in the table, E-Boost provided much better accuracies compared to AdaBoost. The experiments are repeated also for $T = 10$, 15 and 20, and the average accuracies over eleven data sets are presented in Figure 3. The figure show that better ensembles are obtained as the number of classifiers is increased up to 20 which is not the case in AdaBoost.

5 Conclusions

In this study, an evidence theoretic framework is proposed for boosting. Experimental results have shown that the use of local information is highly useful for boosting based classifier ensembles. The proposed approach should also be evaluated for other types of classifiers such as fisher's linear discriminant. Also, the effect of k and τ should be investigated. Our preliminary experiments have shown that the average accuracy increases to 80.82% for $\tau = 0.15$. Also, the performances achieved are observed to be better for smaller k value in some of the data sets. Analysis of the proposed algorithm in terms of these parameters and estimation of their best fitting values are the main topics our current research.

Table 1. Experimental results for $T = 10$ classifiers.

Data set	Base	AdaBoost	E-Boost
monks-1	65.97	65.74	70.83
monks-2	53.47	54.17	64.81
monks-3	80.56	82.64	89.35
wdbc	88.30	88.42	92.28
breast-cancer-w	95.76	95.71	96.54
sonar	65.32	71.77	80.48
ionosphere	74.29	87.05	90.67
liver	55.53	61.75	62.82
heart	61.87	66.37	67.47
phoneme	71.73	71.91	83.73
clouds	75.25	75.27	77.51
Average	*71.64*	*74.62*	*79.68*

References

[1] A. Tsymbal and S. Puuronen. Bagging and boosting with dynamic integration of classifiers. *Principles of Data Mining and Knowledge Discovery, Proceedings of PKDD 2000*, pages 116–125, 2000.

[2] T. Denæux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5):804–813, 1995.

[3] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[4] P. Smets and R. Kennes. The transferrable belief model. *Artificial Intelligence*, 66:191–234, 1994.

[5] J. François, Y. Grandvalet, T. Denæux, and J. M. Roger. Resample and combine: an approach to improving uncertainty representation in evidential pattern classification. *Information Fusion*, 4:75–85, 2003.

[6] R. P. W. Duin. PRTOOLS (version 4). A Matlab toolbox for pattern recognition. *Pattern Recognition Group, Delft University, Netherlands*, 2004.