NONLINEAR DECISION FUNCTION IN SPEAKER VERIFICATION USING A CLASSIFIER ENSEMBLE

Hakan Altınçay and Cem Ergün

Advanced Technology Research and Development Institute, Eastern Mediterranean University
Gazimağusa, KKTC, Mersin 10 Turkey, phone: +90 392 6302842, fax: +90 392 3650711
email: {hakan.altincay,cem.ergun}@emu.edu.tr, web: www.cmpe.emu.edu.tr/altincay

ABSTRACT

The decision rule in speaker verification systems depends on a linear Bayes decision boundary which can be controlled with a threshold. In this paper, the use of complex and nonlinear boundary based decision making is explored which can be achieved using multiple classifier approach. The potential problems in applying such techniques in speaker verification are specified together with some candidate solutions. Then, a well known boosting technique called AdaBoost which is effective in creating an ensemble of classifiers is described. Experiments conducted on NIST99 speaker verification corpus has shown that nonlinear boundary obtained using AdaBoost provides 9.2% improvement in the equal error rate (EER) compared to the Bayes decision making.

1. INTRODUCTION

Pattern classification research includes efforts in devising highly accurate (or, strong) classifiers and the combination of the available approaches. The latter technique covering a wide range of methods is based on the fusion of either strong or weak classifiers and the resultant combined classifiers are generally referred as the multiple classifier systems. Although the latter approach has gained a lot of interest in the recent years, there are still problems which are common to both of the techniques one of which is the class imbalance problem. In some cases such as speaker or face verification, this problem naturally occurs. Consider a speaker verification (*SV*) experiment where a *post-classifier* is to be applied on the verification output scores for optimal decision making [1]. This is basically a 2-class classification problem where the target tests and impostor attacks form the classes. In obtaining the training data for the post-classifier, target tests where the tested identity is the same as the claimed is limited to the number of identities involved, Q whereas $Q \times (Q-1)$ impostor tests where the tested identity is different than that of the tested can be obtained. If Q is large which is the case in almost all *SV* experiments, then the ratio of the impostor scores to the targets available for optimizing the decision function is $(Q-1):1$ which poses a critical problem during decision function optimization.

A generally accepted fact which is also observed in other research domains such as text classification is that, in the case of class imbalance, a classifier may provide much less accuracy for the minority classes (targets in *SV* domain) having much less amount of training data. Several explanations are already available in the literature. For instance, in Ref. [2] it is argued that this is mainly due to the fact that the a priori probabilities bias the learning procedure in favor of the majority class (impostors in *SV* domain). Also, due to the insufficiency of the training data available, the minority class models may not be accurate enough. There are several approaches proposed to deal with the imbalance problem. For instance, *over-sampling* the minority class to make its training data set size same as the majority class or *under-sampling* the majority class. In general, over-sampling is implemented by inserting replicas of the available data points and under-sampling is implemented by taking into account a random subset from the majority class.

AdaBoost algorithm is an iterative multiple classifier system development tool which is shown to provide improved classification accuracies for many different data sets compared to the best individual classifier. In each iteration, a new classifier is trained on a subset of the training data where the weight of each training sample is taken into account in this process. A training sample with a high weight is more likely to be included in the subset than another sample with less weight. Moreover, more than one replica of a training sample may be selected. The selection procedure is generally implemented in the form of roulette wheel selection. At the end of each iteration, the weights of misclassified samples are increased and decreased for those correctly classified. A natural question to ask at this stage is whether the difficulty based sample selection mechanism in AdaBoost may help to avoid the class imbalance problem or not.

In this paper, the behavior of the AdaBoost algorithm in the class imbalance case is initially investigated and the effectiveness of the sampling techniques are verified on a small data set. Then, the AdaBoost algorithm is used together with data sampling to implement a post-classifier to improve the verification decisions. Experiments conducted on the NIST99 speaker verification corpus have shown that 9.2% improvement is achieved in the equal error rate (EER) compared to the Bayes test on the original output scores.

2. ADABOOST IN CLASS IMBALANCE

The AdaBoost (**Ada**ptive **Boost**ing) algorithm is an ensemble creation algorithm which was originally introduced in [3]. 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 of the next classifier. The algorithm summarized in Figure 1. $d_m(n)$ denotes the weight of the nth training sample in $S = \{(x_n, y_n)\}$, $n = 1, \ldots, N$ which is initialized to $1/N$ and C denotes the classifier ensemble where C_m is the classifier obtained at the mth iteration. At the end of each iteration, the weights of the samples that are correctly classified (misclassified) by the new classifier are decreased (increased). In-

1. for $m = 1, ..., M$

1.1 Build classifier C_m using sample set S_m from S using distribution d_m.

1.2 Compute the weighted error using $\varepsilon_m = \sum_{n=1}^{N} d_m(n)(1 - q_{n,m})$ where $q_{n,m} = 1$ if x_n is correctly classified by C_m and zero otherwise.

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

$$d_{m+1}(n) = \frac{d_m(n)}{Z_m} \begin{cases} e^{-\alpha_m} & \text{if } C_m(x_n) = y_n \\ e^{\alpha_m} & \text{if } C_m(x_n) \neq y_n \end{cases}$$

where Z_m is a normalization factor so that d_{m+1} is a distribution.

2. The joint output of the classifier ensemble is computed using

$$C(x) = \sum_{m=1}^{M} \alpha_m C_m(x).$$

Figure 1: The AdaBoost algorithm.

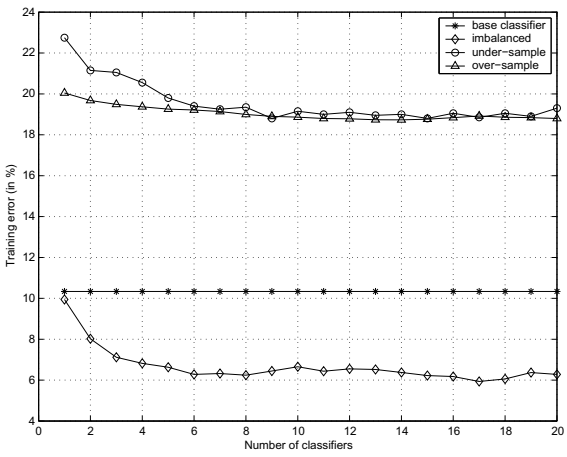

Figure 2: Training error for different number of base classifiers.

creasing the weight of a misclassified sample corresponds to increasing the probability of its inclusion in the training set of the following classifier, probably more than once if its weight is high enough.

In order to evaluate the AdaBoost algorithm in class imbalance case, experiments are conducted on the "phoneme" data set in ELENA database which involves 3818 and 1586 samples respectively for the first and second class. 2500 and 100 samples are randomly selected for training providing an imbalance ratio of 25 : 1. 1300 samples from each class are used for testing. In the under-sampling case, 100 training samples are selected from the first class to be considered for model training whereas in the over-sampling case, the training samples of the second class are replicated for 24 times so that both classes have the same amount of training data. The experiments are repeated for 10 times and the results are averaged. A quadratic discriminant classifier (QDC) from the PRTOOLS toolbox for MATLAB is used as the base classifier, C_m [4]. In the experiments, the number of classifiers in the ensemble is set as $M = 20$.

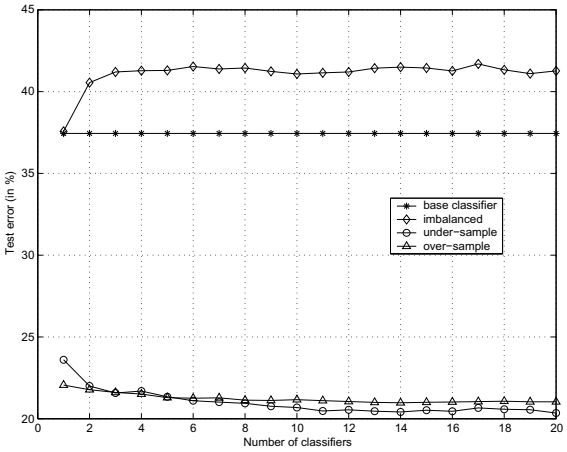

Figure 3: Test error for different number of base classifiers.

Figure 2 illustrates the training error achieved as a function of the classifiers in the ensemble. As seen in the figure, the training performance in the case of imbalanced class case is much better than the sampling based approaches. However, it is evident from Figure 3 that this is mainly due to training inaccurate models that almost always predict the majority class. The poor test performance of AdaBoost indicates that the algorithm is not well suited for imbalanced data sets. In fact, due to the roulette wheel selection mechanism, the percentage of the majority class in each subset will be much larger than that of the minority. The test performance achieved by the sampling techniques prove their efficiency also for the AdaBoost algorithm where the under-sampling performance is slightly better than over-sampling.

Having observed its success on the sampled data, the AdaBoost algorithm is used to implement a nonlinear post-classifier for the *SV* problem. The experiments conducted are presented in the following section.

3. SPEAKER VERIFICATION AND EXPERIMENTAL SETUP

In Speaker Verification (*SV*), the aim is to decide whether the tested speech utterance belongs to the claimed identity or it is an impostor [5]. In the state-of-the-art *SV* systems, the output is composed of two likelihood scores where the decision is based on the likelihood ratio obtained as the difference of the log likelihoods of the outputs,

$$\mathscr{L}_{\mathscr{R}} = \mathscr{L}(X|\lambda_i) - \mathscr{L}(X|\lambda_B) \quad (1)$$

where λ_i denotes the claimant model, λ_B denotes the reference model and X denotes the tested utterance. The decision to accept or reject is based on comparing the likelihood ratio to a threshold, Θ such that

$$\mathscr{L}_{\mathscr{R}} \geq \Theta \implies \text{target speaker} \quad (2)$$
$$\mathscr{L}_{\mathscr{R}} < \Theta \implies \text{impostor} \quad (3)$$

Universal Background Models (UBM) are generally used as the reference models where each UBM is a Gaussian Mixture

Table 1: The number of training and test samples for the minority and the majority classes.

class	training samples	test samples
majority	938520	22071
minority	2376	1479

Table 2: The EER's obtained by using a nonlinear decision boundary estimated using AdaBoost algorithm and the relative improvement compared to the Bayes based baseline (all in percentage).

Base classifier	Conservative	Aggressive
QDC	13.87	14.00
Rel. Improv.	9.2	8.4

Model (GMM) having a large number of mixtures trained to represent speaker-independent distribution of the feature vectors [5]. The claimant models, λ_i are also GMMs which are trained using Bayesian adaptation from the UBM. The short-time spectral information extracted from the speech utterances, Mel-frequency cepstral coefficients (MFCC) are used as the feature vectors. Sixteen MFCCs and their Delta's are computed in every 80 samples for the spectral representation of each Hamming windowed speech frame of length 160 samples.

A subset of the corpus is excluded from verification tests and is used for training the reference model. Approximately one hour of speech is used to train a UBM for male and another one hour of speech for training a female UBM. Each UBM involves 1024 mixtures. Then, these UBMs are combined to obtain a single 2048 mixture joint UBM to be used as a reference model. Excluding the speakers used for UBM training, 396 speakers (245 female and 151 male) are considered during the verification experiments. The training data of each speaker is split into 6 equal parts for 6-fold cross validation to obtain the training data for the multiple classifier based decision boundary. During the cross-validation, the speech segment which is not included in the model training and kept outside for validating the models had impostor attacks on all the other speakers. During the testing phase, non-overlapping speech segments of length 10s are used. The target tests and impostor attacks are performed using the standard setup defined for this corpus.

The output scores corresponding to the tested speaker and the joint UBM are the treated as the inputs for the classifier ensemble to be created using AdaBoost. The number of training and test samples for the minority class (target tests) and the majority class (impostor attacks) are given in Table 1. Due to 396 speakers involved in the *SV* experiment, the ratio of impostor to target training samples is high as 395 : 1. This ratio represents a rather high imbalance ratio. However, it naturally occurs in practice for *SV* problem.

Under-sampling is applied on the impostor scores where a subset whose cardinality is equal to that of the training data considered for the target tests is selected. Two different versions of the AdaBoost algorithm are considered, aggressive and conservative. In [6], the given form of the boosting algorithm in Figure 1 is referred as *Aggressive boosting* since the weights of both correctly and incorrectly classified samples are modified. Alternatively, in *Conservative boosting*, either the weights of misclassified samples are increased or the weights of correctly classified samples are decreased. It should be noted that, Kuncheva *et al.* recommend the use of the conservative version according to their experimental results instead of the original form which is named as aggressive in that context. In the conservative implementations in this study, only the weights of correctly classified samples are updated. The over-sampling approach is not considered in our experiments due to the heavily increased computational load after over-sampling. It may also be argued that the over-sampling approach becomes computationally infeasible as the class imbalance ratio increases.

Quadratic discriminant classifier (QDC) is used as the base classifier. In the AdaBoost algorithm, a newly created classifier is included in the ensemble only if $\varepsilon_m \in (0, 0.5)$. If this is not satisfied, sample weights are reset to $1/N$ and the algorithm continues where $N = 2 \times 2376 = 4752$. In the experiments, the total number of classifiers in the ensemble is selected as $M = 20$. The experiments are conducted for 10 times where the experimental results presented in the following section are the average.

It should be noted that, due to the sample weighting structure of the AdaBoost algorithm, the difficult parts of the decision boundary where the two classes overlap are expected to be treated as more important as the iteration increases. This will result in more classifiers focused on the difficult parts providing the ability of achieving a highly complex boundary.

4. RESULTS

The Equal Error Rate (EER) provided by the baseline *SV* system based on a linear Bayes decision boundary is 15.28%. The EER's obtained using the AdaBoost algorithm are presented in Table 2. It should be noted that the EER is obtained from the Detection Error Tradeoff (DET) curve illustrating the tradeoff between the misclassification of an impostor speaker (false alarm) and a target speaker (miss probability). EER corresponds to the operating point on the DET curve where the false alarm is equal to the miss probability. The DET curves corresponding to the baseline *SV* system and multiple classifier approach are given in Figure 4. As seen in Table 2, the technique considered in this paper provided significant improvements in the EER's compared to the baseline system where log-likelihood ratio of the output scores are directly used. The highest improvement is provided by the conservative application of the AdaBoost algorithm which corresponds to 9.2% relative reduction of the EER.

In the simulation experiments, the subset of training samples selected from the impostor scores has the same cardinality as the target scores set. However, there is no guarantee that this is the best choice [7, 8]. In order to investigate this in the problem under consideration, the experiments are repeated for different amount of samples from the impostor attacks. In other words, the number of impostor samples used during training is selected as r (cardinality ratio) times that of the target tests instead of equal cardinalities. The experimental results are presented in Figure 5. As seen in the figure, this is also true in our case. The minimum EER is achieved when the number of samples selected from the impostors during training is six times that of the targets. Moreover, setting

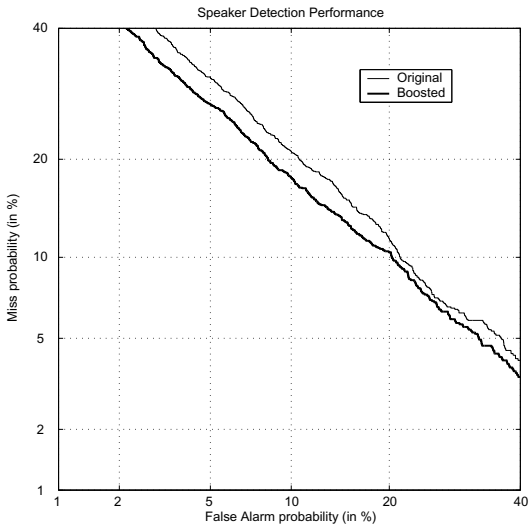

Figure 4: The DET curves obtained for the baseline *SV* system (thin line) and the multiple classifier approach (thick line).

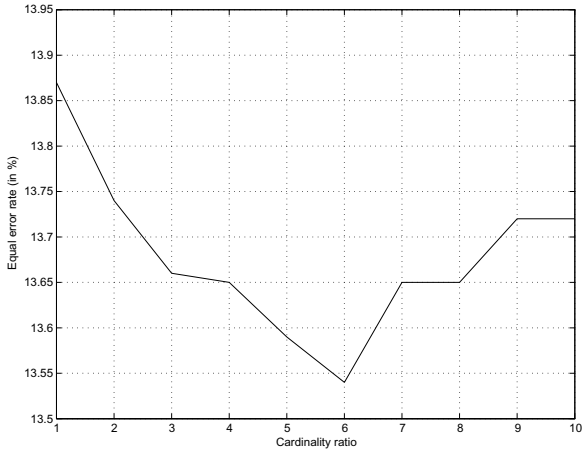

Figure 5: The EER for different cardinality ratios, *r*

the ratio of the cardinalities same as the test data (i.e. 15) provides EER = 13.79% which is not equal to the minimum achievable value. As a matter of fact, the estimation of the optimal value should be further investigated.

5. CONCLUSIONS AND FURTHER WORK

In this study, a multiple classifier system created using AdaBoost algorithm is used to improve the EER of a speaker verification system. The class imbalance problem is addressed and it is observed that the under-sampling approach is a simple but effective method where the AdaBoost algorithm based multiple classifier approach trained using the under-sampled training data provided significant improvement. It is observed that the best results does not correspond to the equal cardinality case which should be further investigated for estimating the best subset.

The under-sampling approach used deserves further attention since it has some drawbacks. Under-sampling discards plenty of available training data which corresponds to loss of available information. Also, it is not guarantied that the subset of data includes sufficient amount of critical samples close in the regions where classes overlap. Moreover, the best test accuracy is not guaranteed for cardinally equal training sets since the class probabilities are highly likely to be different during test phase leading to a larger number of test samples for the impostor attacks. In fact, using more impostor samples during training than targets is shown to provide better test accuracies.

As a further research, it seems reasonable to apply *k*-nearest neighbor rule on the impostor data to identify the regions in the output space where impostor and target outputs overlap. Then, the impostor samples may be sampled according to a probability distribution representing the number of target scores in its *k*-nearest neighbors list. In other words, the impostor scores near the decision boundary may be sampled with a higher probability. An extensive list of available techniques based on the use of neighborhood information in reducing the training set size is available in Ref. [9]. An alternative approach would be applying a clustering technique to identify the naturally available clusters in the impostor data. Then, selecting a set of centroids instead of samples may be an effective method for under-sampling the representative data points in the impostor scores.

REFERENCES

[1] S. Bengio and J. Mariethoz. Learning the decision function for speaker verification. *IEEE-ICASSP Proceedings*, 2001.

[2] G. M. Weiss and F. Provost. The effect of class distribution on classifier learning: An empirical study. *Technical Report ML-TR-44, Department of Computer Science, Rutgers University*, August 2001.

[3] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Second European Conference on Computational Learning Theory*, March 1995.

[4] R. P. W. Duin. PRTOOLS (version 3.0). A Matlab toolbox for pattern recognition. *Pattern Recognition Group, Delft University, Netherlands*, January 2000.

[5] D. A. Reynolds, T. F. Quateri, and R. B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.

[6] L. I.Kuncheva and C. J. Whitaker. Using diversity with three variants of boosting: Aggressive, conservative, and inverse. *Multiple Classifier Systems Workshop, MCS2002.*, 2002.

[7] A. Estabrooks and N. Japkowicz. A mixture-of-experts framework for text classification. In *Proceedings of the Intelligent Data Analysis Conference, IDA*, 2001.

[8] M. C. Monard and G. E. A. P. A. Batista. Learning with Skewed Class Distribution. In J. M. Abe and J. I. da Silva Filho, editors, *Advances in Logic, Artificial Intelligence and Robotics*, pages 173–180. IOS Press, 2002.

[9] D. R. Wilson and T. R. Martinez. Reduction techniques for examplar-based learning algorithms. *Machine Learning*, 38(3):257–286, March 2000.