

Virtual Fusion for Speaker Recognition

Yosef A. Solewicz, Moshe Koppel

Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel

solewicz@013.net, koppel@biu.ac.il

Abstract

This paper presents a simplified post-classification framework for enhancing the performance of a given speaker recognition classifier by means of other “auxiliary” classifiers. We call it *Virtual Fusion*, since the assisting classifiers are used only for training the post-classifier and are not necessary in operating mode. Experiments performed using Nist’04 and ’05 evaluations suggest that the proposed technique is able to consistently improve the EER of a typical GMM-cepstrum classifier by up to 15%.

Index Terms: speaker recognition, fusion, post-classifier.

1. Introduction

Acoustic classifiers are the most accepted classifiers in speaker recognition tasks. Nevertheless, recently, fusion of high-level classifiers has proven to substantially extend the acoustic classifier capabilities [1]. In essence, the fusion process pools the skills of a variety of classifiers, preferably weakly correlated, thus reducing the noise in the inference process.

Ordinary fusion schemes normally attempt to obtain an optimum weighting set for combining the outputs of the several classifiers. The weights are basically a function of the efficiency of the constituent classifiers, their output correlation and the error criterion used. They can be efficiently estimated by means of some machine learning technique such as SVMs, neural networks or even a simple perceptron layer.

In fact, keeping a fixed set of coefficients in order to weight the classifiers’ outputs for each trial is not an optimal procedure. Distinct trials are characterized by different amounts of noise, channel mismatch, speaker mood, etc. Moreover, distinct classifiers are unevenly affected by the diverse sources of noise. Therefore, it would be desirable to consider these artifacts and the classifier strengths when weighting their outputs and not indiscriminately use the same weighting set for all trials. Indeed, lately, some fusion schemes that consider the trial characteristics were proposed [2, 3], and show improvement over ordinary fusion. Similarly, instead of adapting the weights to the trial idiosyncrasies, the corresponding bias reflected in the fused score can be hopefully eliminated. We have recently proposed ABIE (Automatic Bias Identification and Elimination), a post-classification framework that attempts to compensate the bias in classification scores due to a variety of artifacts found the trial utterances. This framework addresses mismatch independently from specific systems or feature sets. It is a general score compensation technique, which uses explicit

feature-level information, and was successfully applied in a variety of low- and high-level speaker recognition systems [4]. In fact, ABIE can be generalized to be jointly applied in a variety of classifiers to be fused [5]. In this case, the post-classifier attempts to learn the relative performance of the individual classifiers on specific trials. In this paper, we focus on a particular application of the post-classification framework for fusion, in which our objective is to optimize a single classifier but additional classifiers are used for training only. We call this procedure *virtual fusion*.

The paper is organized as follows. In Section 2 we review the original ABIE post-classification framework and its application to virtual fusion. A proposal for improving and simplifying this framework is presented in Section 3. In Section 4 we describe experiments performed to validate our assumptions. Concluding remarks are finally discussed in Section 5.

2. ABIE overview

2.1. Basic ABIE

The basic ABIE framework introduced in [4] relies on a post-classifier to reduce bias embedded in scores produced by some speaker recognition system. The post-classifier initially learns a given speaker recognition system’s flaws, trying to correlate erroneous trials of the recognition system with corresponding side-information that presumably reflects the causes of the errors. Each speaker recognition trial is characterized by its side-information vector, as explained later. Once trained, ABIE should be able to predict whether a recognition error is expected given the side-information extracted from the training and testing utterances of a new trial. The post-classifier is trained to output a positive score when a false negative error is expected and a negative score for a false positive error. Its output is then used to correct the correspondent recognition score of this trial:

$$S'_n = S_n + k.T_n \quad (1)$$

where S and S' respectively represent the original and corrected recognition scores of trial n ; T corresponds to the post-classifier’s output given the side-information of trial n , and k is constant.

The side-information that is used for characterizing utterance conditions for purposes of eliminating bias should be orthogonal to the speech features used by the speaker recognition system. While the latter should maximize recognition performance, side-information is supposed to reflect the environment in which an utterance was recorded and should encompass a variety of factors that presumably could be a cause of bias in the specific recognition system.

A side-information vector is obtained for each trial by concatenating sums and absolute differences of time-averaged attributes estimated from train and test utterances, as follows:

This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors’ views.

- Absolute differences between train and test mean cepstral parameters (19 components). (Indicates channel mismatch between both utterances.)
- Absolute differences between train and test standard deviation of the cepstral parameters (19 components). (Indicates noise level mismatch between both utterances.)
- Sum of train and test standard deviation of the cepstral parameters (19 components).
- Absolute differences of: mean pitch, pitch standard deviation, “rate of speech” (zero-crossing of 1st cepstral coefficient), between train and test (3 components). (Indicates mismatch in speaking style.)
- Sum of: mean pitch, pitch standard deviation, “rate of speech”, between train and test (3 components).

2.2. Fusion and virtual fusion

The basic ABIE framework can be generalized to operate as a post-classifier in a fusion scheme rather than in a single system [5]. Thus, rather than learning to eliminate bias in a given classifier, the post-classifier correction is based on the comparative performance of distinct classifiers on specific speaker recognition trials.

The motivation behind this approach is as follows. In general, speaker recognition fusion relies primarily on the more accurate low-level acoustic classifiers and to a lesser degree on high-level linguistic classifiers. Consequently, since low-level classifiers dominate the fused score, poor performance of these classifiers would lead to poor classification regardless of the performance of the high-level classifiers. On the other hand, the highly weighted low-level classifiers might overshadow particularly good performance of high-level classifiers in certain trials.

In the ABIE framework for fusion, the post-classifier can be trained alternatively in one of two operating modes, attempting to spot trials that would either lead to poor low-level classifier performance or obtain good high-level performance. In particular, in *Mode I*, we attempt to spot the trials poorly classified by the low-level classifiers. This is accomplished training the post-classifier to discriminate between trials poorly classified by the low-level classifiers vs. trials poorly classified by the high-level classifiers. Similarly, in *Mode II*, we attempt to spot the trials well classified by the high-level classifiers. This is accomplished training the post-classifier to discriminate between trials well classified by the high-level classifiers vs. trials well classified by the low-level classifiers. Again, each trial is represented by its corresponding side-information vector as described above.

In this scheme, the available systems must be assigned to either a low- or high- level partition. For our purposes, we use a single “low-level” classifier consisting of a GMM-cepstrum system (the base classifier) and a variety of “high-level” systems to be described below.

We have observed that best results are obtained when the post-classifier is trained exclusively using speaker recognition target trials and discarding impostor trials. (The use of target trials, in which the same speaker appears in train and test utterances, more effectively neutralizes speaker specific variability and highlights other sources of bias.) Therefore, for training the post-classifier, we denote by ‘well classified’, target trials having high scores. Correspondingly, ‘poorly classified’ trials are target trials having low scores.

The post-classifier is trained such that when operating in Mode I, it should output a positive value whenever a base

classifier flaw is predicted. On the other hand, if it operates in Mode II, it should output positive values when the trial is supposed to be well classified by the high-level classifiers. The post-classifier outputs are then used to correct the recognition scores \mathcal{S} , in (1).

Note that these scores can be the fused recognition scores or simply the base classifier’s scores. In the former possibility, we initially fuse the available classifiers in order to obtain the scores \mathcal{S} (and later use the individual classifiers to train the post-classifier). In this paper, we will focus on the latter case (virtual fusion), in which we are required to obtain scores for the high-level partition classifiers only for training the post-classifier and not for actual operation, since we are not fusing them.

3. Parameter Optimization

In order to train the post-classifier, we must determine which trials of both low- and high-level partitions will be used for training. In Mode I, we use the poorly classified trials of both partitions in order to spot the low-level classifier’s flaws and conversely in Mode II, well classified trials of both partitions are used to spot the high-level classifiers’ hits. Actually, the percentage of target trials considered, either poorly (Mode I) or well (Mode II) classified must be optimized. Few such examples could lead to poor training, while an excessive amount of examples might comprise also typical trials, thus forming a noisy training set. In addition, we must choose a proper value for k in (1). The necessary percentages and k can be optimized using hold-out data [4] or parametrically [5].

However, another approach can be considered in which we simply fix a priori the required parameters. This approach might not be optimal for distinct high-level classifier partitions. Nevertheless, it has the advantage of avoiding tricky parameter optimizations.

Moreover, we would like to unify the two proposed operation modes, thus hopefully increasing performance. We therefore introduce a preceding classification layer, responsible for discriminating between suitable and unsuitable trials, based on their side-information vectors. Thus, at first, the input side-information vector is classified into either a potentially well (suitable) or poorly classified (unsuitable) trial. In case the trial is considered to be inappropriate, a post-classifier operating in Mode I will try to predict whether this unsuitability is due to a base-classifier flaw. On the other hand, if the trial is found to be suitable, another post-classifier operating in Mode II will attempt to predict if it is especially suitable for the high-level classifiers. This scheme is illustrated in Figure 1.

Nevertheless, simultaneously using two post-classifiers (Modes I and II) results in more parameters to be jointly optimized and increasing the number of free parameters in the post-classification scheme turns the optimization process more complex. In fact, in order to train both post-classifiers, we must determine the percentages of well and poorly classified trials from both the base classifier and those forming the high-level partition. Let $\mathbf{p}_b(w)$ and $\mathbf{p}_b(p)$ be the percentages of well and poorly classified trials by the base classifier. Similarly, let $\mathbf{p}_h(w)$ and $\mathbf{p}_h(p)$ be the percentages of well and poorly classified trials of each of the high-level partition classifiers. We now propose a simplification in order to reduce the number of percentage values to be determined as follows.

In order to reduce the number of parameters to be optimized, we initially collapse $\mathbf{p}_h(w)$ and $\mathbf{p}_h(p)$ into a single parameter, \mathbf{p}_h . Thus, we use the same percentage of well and

poorly classified trials by each of the high-level partition classifiers in order to train the post-classifiers.

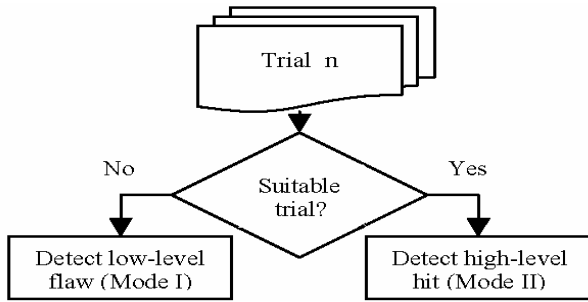


Figure 1: Schematic view of the whole post-classification process.

Moreover, we have observed that in general, the required percentage of the base classifier trials and the parameter k are relatively constant for a variety of high-level classifier partitions. Therefore, instead of performing optimization, we empirically fix constant values for k , $\mathbf{p}_b(w)$ and $\mathbf{p}_b(p)$. The reason for not fixing also \mathbf{p}_h is that it depends on the type of classifiers found in the high-level partition. Specifically, \mathbf{p}_h depends on the degree of correlation (measured by the correlation coefficient) among these classifiers and the base classifier. As we shall see later, in case the degree of correlation is high, more trials are required when training the post-classifiers, since several trials are common to both groups. (i.e., the well or poorly classified trials will be roughly the same for the base and the high-level classifiers). Conversely, low correlated classifiers will lead to more dissimilar trials between the two groups, alleviating the post-classifiers training.

In summary, we are left with a single free parameter to be determined, \mathbf{p}_h . As explained, this parameter should be proportional to the degree of correlation among the base and high-level classifiers. In fact, for a given \mathbf{p}_h , the total amount of dissimilar trials in both groups, will be mainly determined by the high-level classifier showing the least correlation with the base classifier. Therefore, we select \mathbf{p}_h to follow the least correlated classifier found in the high-level partition, so that a high correlation degree requires a high \mathbf{p}_h . These assumptions in fact make unnecessary any automatic parameter optimizations and are validated in the next section.

4. Experiments

In order to validate the proposed technique, experiments were conducted using the NIST'04 evaluation as a development set and NIST'05 evaluation as a test set. We used recognition scores made available by SRI for seven different systems [6]. The systems span several speaker recognition layers. Roughly speaking, the systems can be categorized either into acoustic (Systems 1 to 3) or stylistic (Systems 4 to 7) oriented. The acoustic systems are based on derivations of cepstral features and components of the maximum likelihood linear regression (MLLR) transforms. By contrast, the stylistic systems explore counts and duration of words and other prosodic features extracted over automatically estimated syllables (SNERF). Table 1 lists the systems, their performance in Nist'05 in terms of Equal Error Rate (EER) and in addition the degree of correlation between our base classifier, the Cepstral-GMM and the other systems.

Table 1. Systems performance and correlation with the base classifier (System 1).

System	Description	EER (%)	Correlation (%)
1	Cepstral GMM	7.26	100.00
2	Cepstral SVM	7.26	91.72
3	MLLR transform SVM	10.34	79.42
4	SNERF	14.11	55.42
5	State Duration	15.38	52.57
6	Word Duration	19.27	32.51
7	Word N-gram SVM	24.56	25.81

In these experiments, our goal is to improve the performance of the base classifier through virtual fusion. The full post-classification scheme comprises a preliminary layer realized by a simple Bayesian classifier followed by two linear SVM classifiers as illustrated in Figure 1.

The first layer determines whether a certain trial is suitable for classification, given its side-information vector. This classifier is trained with trials well classified by the base classifier vs. trials poorly classified by this classifier. (Trials from the high-level classifiers could be also considered but it seems to be superfluous.) Actually, we use about 10-20% of the highest-score target trials and the same amount of the least-scored target trials. Multivariate Gaussian probability density functions are estimated for both hypotheses, using the correspondent side-information vectors.

In testing mode, the preliminary classifier performs a likelihood test and directs each trial to the proper post-classifier. In case of an expected unsuitable trial, this trial will be further classified by the post-classifier operating in Mode I. Otherwise, if the trial is likely to be suitable, it is directed to the post-classifier operating in Mode II.

As noted before, those two post-classifiers are implemented by linear SVMs. The percentages $\mathbf{p}_b(w)$ and $\mathbf{p}_b(p)$ are empirically set to 10% and 30%. The value \mathbf{p}_h , supposedly being a function of the least valued correlation coefficient between the base classifier and the classifiers comprising the high-level partition (see Table 1) is empirically set to 10%, for low correlated classifiers (specifically, Systems 6 and 7); 20%, for moderately correlated classifiers (Systems 4 and 5); and 30%, for highly correlated classifiers (Systems 2 and 3). The constant k is set to 0.25.

The preliminary and the two post-classifiers are trained in Nist'04 evaluation using the above settings. Several high-level classifier partitions are used (arrangements of Systems 2 to 7 in Table 1) and the correspondent EERs obtained in Nist'05 evaluation are depicted in Figure 2.

We compared the results obtained fixing the parameters, with those obtained after $\{\mathbf{p}_b(w)$ and $\mathbf{p}_b(p)$, \mathbf{p}_h , $k\}$ optimization using hold-out data, as noted in Section 3. We found that while specific parameter optimization attained better results for some high-level classifier partitions, there are other many cases where this is not the case. Thus, keeping a fixed set of parameters is a cheap and not-far-from-optimal alternative.

In addition, we observed that the proposed scheme is relatively stable for $\{\mathbf{p}_b(w)$ and $\mathbf{p}_b(p)$, \mathbf{p}_h , $k\}$ settings close to those used. In special, we have confirmed our assumptions about the dependency of \mathbf{p}_h on the correlation degree between

the base classifier and those (in particular the least correlated) comprising the high-level partition. Even though increasing p_h fairly compensates for excessive correlation degrees, as a rule, best configurations are those including less correlated classifiers. The inclusion of other typically low-level classifiers (highly correlated with the base classifier, like Systems 2 and 3) in the high-level partition seems to burden the post-classifier training, since similar trials will be present simultaneously as positive and negative training examples.

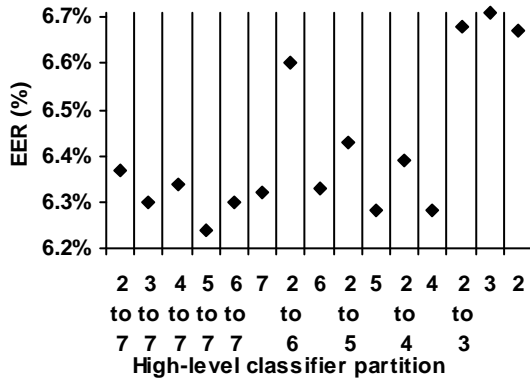


Figure 2: Post-classification performance for distinct high-level classifier partitions.

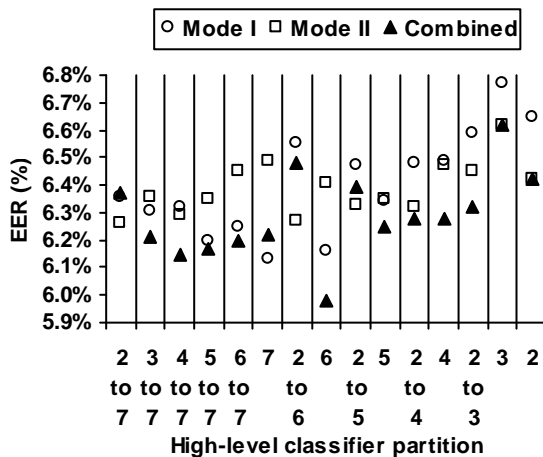


Figure 3: Optimal EERs for distinct high-level classifier partitions and different post-classifier training modes.

We further wished to assess the improvements obtained by the preliminary classification layer introduced to merge Modes I and II. As explained earlier, this layer should switch between the two post-classifiers according to the suitability of the incoming trials. (Interestingly, the main feature exploited by this classifier is the noise mismatch in low spectrum and secondly, mismatch in mean pitch and speech rate. The main features used by the post-classifiers are mainly noise and channel mismatch in the low spectrum.) In order to obtain a fair comparison and neutralize the influences of properly setting $\{p_b(w)$ and $p_b(p)$, p_h , $k\}$, we optimized these parameters a posteriori for each of the high-level classifier partitions evaluated. The results obtained are depicted in Figure 3.

It can be observed that in general, individually, Mode I is more appropriate to be used when the available high-level classifiers are relatively weakly correlated with the base classifier. On the other hand, Mode II seems to be a better option, when the available high-level classifiers are quite correlated with the base classifier. Moreover, the proposed scheme merging both modes is apparently successful in exploiting the advantages of Modes I and II, outperforming both modes individually in a wide range of high-level classifier partitions.

5. Conclusions

We have presented a simplified post-classification scheme able to improve the performances of speaker recognition systems. This framework requires auxiliary speaker recognition systems for training the post-classifiers. In case the auxiliary systems are available at operational mode, the framework can be applied on the fused systems' scores. If these systems are available only off-line, the framework can be applied simply on the base classifier, performing virtual fusion. Proper auxiliary systems are those poorly correlated with the base classifier, decreasing EER in around 15%. We intend to further optimize this framework and in particular formalize the parameters setting procedures.

6. Acknowledgements

The authors would like to thank SRI International for kindly providing their evaluation scores.

7. References

- [1] J. Campbell, D. Reynolds, and R. Dunn. "Fusing high- and low-level features for speaker recognition". In Proc. 8th European Conference on Speech Communication and Technology, Geneva, Switzerland, pp. 2665-2668, 2003.
- [2] Y. Solewicz and M. Koppel, "Enhanced Fusion Methods for Speaker Verification", Proceedings of the 9th International Conference "Speech and Computer", St. Petersburg, Russia, pp. 388-392, 2004.
- [3] L. Ferrer, K. Sonmez, S. Kajarekar, "Class-dependent score combination for speaker recognition". In Proc. 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, pp. 2173-2176, 2005.
- [4] Y. Solewicz and M. Koppel, "Automatically correcting bias in speaker recognition systems", In Proc. 16th IEEE Workshop on Machine Learning for Signal Processing, Maynooth, Ireland, pp. 187-191, 2006.
- [5] Y. Solewicz and M. Koppel, "Using Post-Classifiers to Enhance Fusion of Low- and High-Level Speaker Recognition", To appear in IEEE Transactions on Audio, Speech and Language Processing.
- [6] L. Ferrer, E. Shriberg, S. S. Kajarekar, A. Stolcke, K. Sonmez, A. Venkataraman, and H. Bratt. "The Contribution of Cepstral and Stylistic Features to SRI's 2005 Nist Speaker Recognition Evaluation System", Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, pp. 101-104, 2006.