Using Post-Classifiers to Enhance Fusion of Low- and High-Level Speaker Recognition

Yosef A. Solewicz and Moshe Koppel

***Abstract*—This paper proposes a method for automatic correction of bias in speaker recognition systems, especially fusion-based systems. The method is based on a post-classifier which learns the relative performance obtained by the constituent systems in key trials, given the training and testing conditions in which they occurred. These conditions generally reflect train/test mismatch in factors such as channel, noise, speaker stress, etc. Results obtained with several state-of-the-art systems showed up to 20% decrease in EER compared to ordinary fusion in the NIST'05 Speaker Recognition Evaluation.**

***Index Terms*—Fusion, machine learning, post-classification, speaker recognition.**

I. Introduction

AUTOMATIC speaker recognition is the task of person authentication by an automatic analysis of the person's voice. Speaker-specific characteristics are due to differences in both physiological (low-level) and behavioral (high-level) aspects of the speech production system. While low-level features reflect anatomical constraints, high-level features also track behavioral trends on speakers. People tend to prefer certain words and avoid others. The same occurs with conversational and intonational patterns. Traditionally, speaker recognition has been based on the former factor, essentially by means of acoustic features extracted from the speech signal. Acoustic speaker recognition is more robust and requires less data for training and testing. In recent years though, the advent of extended databases, powerful machine learning algorithms and abundant computational power have brought about a new reality. Fusion of several systems exploring behavioral trends has been shown to significantly boost recognition performance [1]–[3].

Ordinary fusion commonly uses some machine learning paradigm in order to find a set of fixed weights for base classifiers' outputs, which minimize some error criterion on a development set. This approach is not optimal in the sense that weights are fixed and optimized for a specific composition of different "types" of trials in the development set and will not necessarily perform well for a different balance of trials. Moreover, whatever the learning algorithm applied, ordinary fusion does not fully exploit the synergy of classifier combination. It is known that the several feature sets possess individual strengths, each one being more appropriate to a specific environment. In this sense, a blind classifier combination, ignoring the kind of data being classified is far from an optimal procedure.

Humans, on the other hand, can activate different levels of speech perception according to specific circumstances, by having certain processing layers compensate for others affected by noise. Utterance length, background noise, channel, and speaker emotional state are some of the parameters which might dictate the form by which one will perform the recognition process.

Recent work addresses this limitation by assigning different fusion schemes for distinct classes of trials [4]–[6]. In these schemes, the type and degree of distortion found in the speech sample to be classified is implicitly or explicitly integrated into the classification task. Thus, for example, although acoustic features are generally far superior to all other feature types, there are circumstances under which more weight should be given to lexical features.

In this paper, we describe a system called Automated Bias Identification and Elimination (ABIE), for optimally weighting constituent systems in a fusion-based speaker recognition system according to a variety of utterance characteristics. The basic system, introduced in [7], explores utterance characteristics within isolated systems in order to reduce train/test mismatch effects. In this paper, we propose a generalization of this framework to be applied in fusion of recognition systems.

Bias in fused systems is a complex process, resulting from the interaction of distinct systems, each one with its own bias artifacts. More specifically, mismatch is caused when models are learned from training utterances recorded under a particular set of conditions, but these models are then applied to testing utterances recorded under a different set of conditions. The distinct recording environments might unevenly distort train/test feature distributions, thus introducing bias in the computed scores and finally causing light to severe degradations in system performance.

Mismatch factors such as transmission channel, recording media, and background noise have been the subject of intense research. In recent years, several techniques for channel and additive noise compensation have been proposed. These compensation techniques are traditionally employed at the feature level [8]–[12], aiming to compensate for the bias introduced in speech features' distributions, or at the score level [13], wherein normalization is performed based on the behavior of scores obtained by a reference population in similar operating conditions.

Recently, model-based compensation techniques have been proposed. They are based on latent factor analysis in order to compensate for channel and speaker variability in Gaussian mixture model (GMM) classifiers [14]. A parallel approach was

Manuscript received February 11, 2007; revised May 13, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Joseph Campbell.

The authors are with the Department of Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel (e-mail: solewicz@013.net; koppel@cs.biu.ac.il).

Digital Object Identifier 10.1109/TASL.2007.903054

also considered for support vector machine (SVM) classifiers [15]. These techniques exploit discrepancies between different conversations by the same speaker ("intersession variability") in order to isolate extraneous effects, thus requiring multiple sessions from distinct speakers for training.

Until recently, mismatch was seen exclusively as an acoustic issue. Nevertheless, since higher order feature levels have been increasingly explored for speaker verification, proper compensation schemes should be developed for these novel feature sets.

The basic ABIE framework addresses mismatch independently of specific systems or feature sets. It is a general score compensation technique, which uses explicit feature-level information, from both high-level and traditional low-level speech features. The core of ABIE is a post-classifier that learns the errors of any recognition system, based on side-information reflecting the environment in which the utterances were recorded. The type of error (false alarm or false reject) of each erroneous trial is associated with side-information extracted from the training and testing utterances comprising this trial. Once trained, ABIE should be able to predict whether a recognition error is expected given the side-information of the training and testing utterances of some trial. ABIE's outcome can then be used either to redefine a system design or to correct its scores.

This approach offers several advantages. First of all, ABIE can be easily overlaid on those methods previously described, yielding further improvement in accuracy, with a very low computational overload in operating mode, compared to current compensation techniques. Although operating on the score level, ABIE provides an explicit insight into bias sources due to mismatched conditions in training and testing utterances. Moreover, the user openly selects the side-information attributes to be used. By contrast, in intersession variability methods, variability is normally viewed as a whole, and individual bias sources are not explicitly explored. Furthermore, in those feature [12] or score normalization [13] techniques which do use side-information, attributes are explored in isolation, and their mutual influence on score bias is not fully explored as in ABIE.

This framework was evaluated on a variety of low- and high-level speaker recognition systems, considerably improving individual performance [7].

In this paper, we generalize the ABIE framework capabilities so as to support fusion of classifiers as well. Thus, as opposed to other compensation methods, ABIE allows to simultaneously model mismatch among several subsystems. This is accomplished by considering the relative performance of individual classifiers on certain trials, given their characteristics, in order to compensate for inadequacies inherent to ordinary fusion techniques. In fact, we show that this approach can be successfully employed on single systems too.

The outline of this paper is as follows. In Section II, we offer a brief overview of the basic ABIE system. In Section III, we present its fusion version and describe the training procedure developed for this purpose. In Section IV, we report experiments performed and analysis of results. "Virtual fusion," the application of this framework to single systems is shown in Section V. Finally, Section VI is dedicated to a concluding discussion and suggestions for future work.

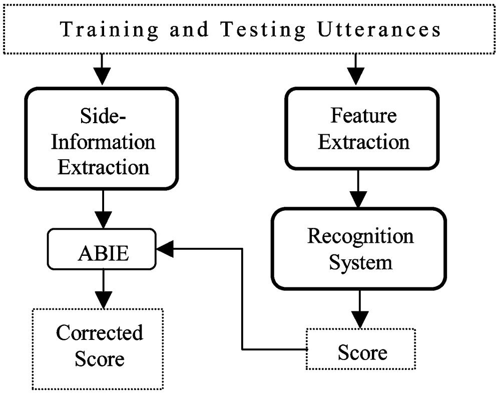


Fig. 1. Schematic representation of ABIE.

II. Basic ABIE Overview

The basic ABIE is a framework based on a post-classifier that learns a given speaker recognition system's flaws. It tries to correlate erroneous trials of the recognition system with corresponding side-information that presumably reflects the causes of the errors. Then, in operating mode, given side-information about a particular training/testing trial, the post-classifier attempts to correct the score obtained by the speaker recognition system. The ABIE operation scheme is summarized in Fig. 1 and a brief description is given below (more details can be found in [7]).

A. Training Procedure

1) Given some speaker recognition system, obtain several target and impostor scores.
2) Extract side-information (as explained below) from each of the trials in step 1).
3) Sort the scores of each distribution and label as "+1" a percentage $\mathrm{p_t}$ of the lowest target scores (false rejects) and as "−1" a percentage $\mathrm{p_i}$ of the highest impostor scores (false accepts).
4) Train a classifier to discriminate between the $\mathrm{p_t}$ and $\mathrm{p_i}$ examples, given the respective side-information vectors.
5) Apply the trained classifier to some set of speaker recognition trials $X_{1,\ldots,n}$ (i.e., input the correspondent side-information vectors) and obtain scores $T_{1,\ldots,n}$.
6) Perform score correction adding the classifier outputs $T_{1,\ldots,n}$, scaled by a constant k to the correspondent recognition system's scores $\mathrm{S}_{1,\ldots,n} : \mathrm{S}'_{1\ldots n} = \mathrm{S}_{1\ldots n} + k.T_{1\ldots n}$.
7) Iterate steps 3)–6) above optimizing the triple $\{\mathrm{p}_t, \mathrm{p_i}, \mathrm{k}\}$ so as to minimize the corrected recognition error on trials $X_{1,\ldots,n}$.
8) Retrain the classifier using the optimized $\{\mathrm{p_t}, \mathrm{p_i}, \mathrm{k}\}$ and employ it as a post-classification layer following the speaker recognition system.

B. Side Information

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.

The side-information vector used in the present experiments attempts to cover a rough estimate of channel, noise, and prosody aspects found both in train and test utterances. We initially extract several speech parameters (63 attributes as described below) at a frame step of 20 ms for both the train and test utterances. These parameters are then averaged in time for each utterance. Finally, we obtain sums and absolute differences of the averaged attributes that are then concatenated into a single side-information vector per trial as follows.

- Absolute differences between train and test mean cepstral parameters (19 components). This is an indication of channel mismatch between both utterances [10].
- Sum of train and test standard deviation of the cepstral parameters (19 components). This is an indication of the amount of additive noise in the trial [10]. (For simplicity, we denote hereafter cepstrum standard deviation by "quality.")
- Absolute differences between train and test standard deviation of the cepstral parameters (19 components). This is an indication of quality mismatch between both utterances [10].
- Absolute differences of: mean pitch, pitch standard deviation, "rate of speech" (zero-crossing of first cepstral coefficient), between train and test utterances (three components). This would roughly reflect mismatch in higher-levels: speaking style, speaker mood, etc.
- Sums of: mean pitch, pitch standard deviation, "rate of speech," between train and test utterances (three components).

III. ABIE for Fusion

In this section, we extend the basic ABIE framework to operate as a post-classifier in a fusion scheme rather than in a single system. Thus, rather than learning to eliminate bias for a given classifier, we propose now to apply a correction 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. Since low-level classifiers dominate the fused score in ordinary fusion, poor performance of these classifiers in specific trials would lead to poor classification regardless of the performance of the high-level classifiers. In addition, the highly weighted low-level classifiers might overshadow trials in which high-level classifiers show particularly good performance. Hence, it would be desirable to spot trials that would either lead to poor low-level classifier performance or obtain good high-level performance. These trials should have their scores compensated; specifically, the scores should be increased in case of positive trials or decreased in case of negative trials.

We therefore propose two alternative training modes for the post-classifier. 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 versus 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 versus trials well classified by the low-level classifiera.

We initially denote by "well classified," either target trials having high scores or impostor trials having low scores. Correspondingly, "poorly classified" trials are either target trials having low scores or impostor trials with high scores. In a second step, we must consider which type of trials (target, impostor, or both) will be used to train the post-classifier in the two modes mentioned above. Furthermore, we must determine the optimum categorization concerning low- and high-level classifiers, i.e., how to group all available classifiers into two classes, either low- or high-level. Although low-level is typically associated with acoustic feature sets and high-level with linguistic parameters, it is not necessarily the case that this categorization is strictly suitable for the proposed framework.

Several preliminary experiments were performed in order to address the above questions. These experiments seem to indicate that the best option is to train the post-classifier exclusively on target trials. This somewhat counter-intuitive conclusion can be explained as follows. Side-information vectors as defined above are intended to simply reflect the context in which model and test utterances were obtained. Nevertheless, these vectors inevitably also include an undesired speaker-specific component. Since they address mismatch between model and test time-averaged attributes, target trials, in which model and test utterances pertain to the same speaker, are more effective in neutralizing speaker-specific bias and emphasizing exclusively contextual mismatch.

Furthermore, it was observed that some flexibility is warranted in grouping the classifiers into low- or high-level classes for training. In particular, a good configuration was to consider the best acoustic classifier as the unique low-level classifier and all the others as high level-classifiers. In this sense, we could pose the current challenge as enhancing the performance of a standard acoustic classifier by means of auxiliary classifiers. (Although this sounds almost synonymous with fusion, we show later in Section V that the two approaches can be conceptually dissimilar.)

Other configurations, such as simply discarding the remaining low-level classifiers or grouping the classifiers according to the conventional notion of low- and high-level feature sets, also worked well. We note that for the moment we define the post-classification stage as a two-class problem (low versus high-level), but that in principle, it would be appropriate to generalize the framework to encompass more than two such classes.

A. *Modified Training Procedure*

Training the post-classifier in one of the two modes proposed is similar to the methodology presented in Section II-A, for the basic ABIE framework. The main difference is that now we contrast among either poorly classified (Mode I) or well-classified (Mode II) examples from distinct classifiers and not from false positive and false negative examples pertaining to a single system. Recall that we consider only speaker recognition target trials in order to train the post-classifier. Therefore, we mean by "poorly classified" or "well classified," target trials whose scores are, respectively, low or high. The number of such trials used to train the post-classifier is quantified by percentages of trials with either lowest or highest scores relative to the whole target distribution.

Specifically, in Mode I, the post-classifier is trained to discriminate between side-information vectors of poorly classified trials from low- versus high-level speaker recognition classifiers. The union of all side-information vectors from poorly classified trials belonging to the low-level classifier partition is labeled as "$+1$" and the union of vectors of poorly classified trials belonging to the high-level classifier partition is labeled as "-1." We therefore expect the post-classifier to output a high score for (positive) trials which might be eventually misclassified by the low-level classifiers.

Similarly, in Mode II, we label the side-information vectors of well-classified trials belonging to the high-level partition as "$+1$" and as "-1" the well-classified vectors from the low-level partition. Again, we expect high post-classifier outputs for (positive) trials that are appropriate to high-level classifiers.

The number of well-classified or poorly classified examples used to train the post-classifier in either mode are determined by $\mathrm{p_l}$ and $\mathrm{p_h}$. (These percentages play a similar role as $\mathrm{p_t}$ and $\mathrm{p_i}$ in Section II-A.) The percentage $\mathrm{p_l}$ corresponds to the worst (Mode I) or the best (Mode II) scores of each of the low-level classifiers, and $\mathrm{p_h}$ is analogous for the high-level classifiers. Thus, in Mode I, the positive training examples used by the post-classifier are formed by the union of the $\mathrm{p_l}$ worst examples of each of the low-level classifiers, while the negative examples are formed by the union of the $\mathrm{p_h}$ worst examples of each of the high-level classifiers. Correspondingly, in Mode II, the positive training examples used by the post-classifier are formed by the union of the $\mathrm{p_h}$ best examples of each of the high-level classifiers, while the negative examples are formed by the union of the $\mathrm{p_l}$ best examples of each of the low-level classifiers. In each mode, the set $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ is optimized as detailed in the next subsection.

B. Parametric Optimization

It is straightforward to employ the basic ABIE training procedure (Section II-A) in order to train the modified post-classifier. The set of parameters $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ can be similarly optimized through greedy search. Thus, for each $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$, the corrected scores are explicitly computed according to equations (1a) and (1b) below, and some error measure is estimated between the target and impostor resulting distributions. The optimum set $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ is the one which leads to minimum error. We call this an *explicit* optimization (a discriminative approach), as opposed to a *parametric* optimization (a generative approach). In the parametric approach, instead of explicitly calculating the target and impostor distributions of the corrected scores, we simply estimate parametric representations for these distributions. The complete parametric optimization is performed as follows.

We model the original recognition score distributions (S) and post-classifier outputs (T), as normal random variables. (As we will see below, in the present experiments, the post-classifier is realized by an SVM. It can be observed that, in fact, the scores produced by the SVM closely follow a Gaussian distribution. In case other classifiers are used, some mapping could be performed on the scores in order to shape a normal distribution or, alternatively, they could be approximated by other types of random variable.) In particular, let S_t and S_i represent the target and impostor recognition scores respectively, with means μ_t and μ_i, and variances σ_t^2 and σ_i^2. Correspondingly, T_t and T_i represent the post-classifier outputs for target and impostor matches, with means $\bar{\mu}_t$ and $\bar{\mu}_i$, and variances $\bar{\sigma}_t^2$ and $\bar{\sigma}_i^2$. Thus, the corrected scores of target $(\mathrm{S'_t})$ and impostor $(\mathrm{S'_i})$ trials are given by

$$\mathrm{S}'_t = \mathrm{S_t} + \mathrm{k} \cdot T_{\mathrm{t}} \tag{1a}$$

$$\mathrm{S}'_i = \mathrm{S_i} + \mathrm{k} \cdot T_{\mathrm{i}}. \tag{1b}$$

Recall that k is a constant to be optimized and that T_t and T_i depend on $\{\mathrm{p_h}, \mathrm{p_l}\}$, which define the number of examples used to train the post-classifier. According to our assumptions about S and T, it is easy to see that S'_t and S'_i are also normal random variables with mean μ' and variance σ'^2, respectively, given by

$$\mu'_t = \mu_t + \mathrm{k} \cdot \bar{\mu}_t \tag{2a}$$

$$\mu'_i = \mu_i + \mathrm{k} \cdot \bar{\mu}_i \tag{2b}$$

$$\sigma_t'^2 = \sigma_t^2 + 2 \cdot \mathrm{k} \cdot \mathrm{cov}(\mathrm{S_t}, T_t) + \mathrm{k}^2 \cdot \bar{\sigma}_t^2 \tag{3a}$$

$$\sigma_i'^2 = \sigma_i^2 + 2 \cdot \mathrm{k} \cdot \mathrm{cov}(\mathrm{S}_i, T_i) + \mathrm{k}^2 \cdot \bar{\sigma}_i^2. \tag{3b}$$

The operator cov stands for the covariance.

At this point, the corrected scores S'_t and S'_i, for a given $\{\mathrm{p_h}, \mathrm{p_l}\}$, are represented by two normal score distributions, with means [(2a), (2b)], and variances [(3a), (3b)] being a function of k

$$\mathrm{N}_t \sim N(\mu'_t(\mathrm{k}), \sigma_t'^2(\mathrm{k})) \tag{4a}$$

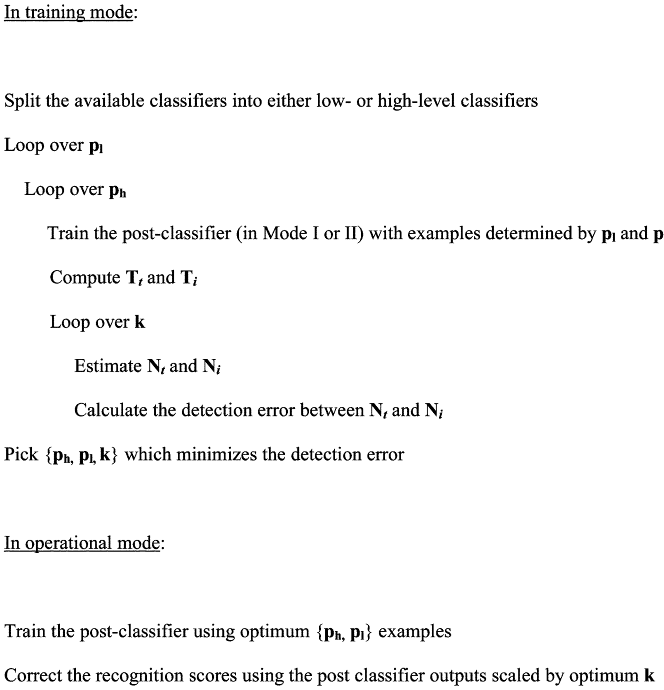
$$\mathrm{N}_i \sim N(\mu'_i(\mathrm{k}), \sigma_i'^2(\mathrm{k})). \tag{4b}$$

Finally, we can analytically find the value of k that minimizes some error criterion between the two distributions [(4a), (4b)]. In short, the complete training process involves the joint optimization of the triple $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ and is performed following the pseudocode in Fig. 2.

IV. 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. These evaluations consist of 10 000–20 000 trials, involving 500+ speakers recorded in a variety of landline/cellular lines [16]. There are no cross-gender trials, and the evaluation balance is about ten impostor trials for each target trial.

We used recognition scores made available by SRI International for seven different systems. The systems span several speaker recognition layers. Roughly speaking, the systems can be categorized either into acoustic (Systems 1 to 3) or stylistic



```
In training mode:

Split the available classifiers into either low- or high-level classifiers
Loop over $p_l$
  Loop over $p_h$
    Train the post-classifier (in Mode I or II) with examples determined by $p_l$ and $p_h$
    Compute $T_t$ and $T_i$
    Loop over $k$
      Estimate $N_t$ and $N_i$
      Calculate the detection error between $N_t$ and $N_i$
Pick $\{p_h, p_l, k\}$ which minimizes the detection error

In operational mode:

Train the post-classifier using optimum $\{p_h, p_l\}$ examples
Correct the recognition scores using the post classifier outputs scaled by optimum $k$
```

Fig. 2. Pseudocode for training and operating the post-classifier.

(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). All these systems use either GMM or SVM classifiers for modeling their respective feature sets, and their brief description can be found in [17]. We include an extra system, which represents the fusion of the above systems in which the respective weights are determined from training data by a linear SVM [18]. Table I lists the systems, their performance in NIST'05 in terms of equal error rate (EER) and in addition the correlation coefficient between the cepstral-GMM (target) scores and those obtained by the other systems. This correlation value can be viewed as a quantitative measure of how much a system can be considered as high-level. The lower the correlation, the more the system behavior diverges from that of the cepstral-GMM which was chosen as the low-level reference system, as explained below. In addition, note that system performances follow the correlation trend and decrease for higher level systems.

In these experiments, our goal is to improve the performance of ordinary fusion used in System 8. We train post-classifiers which are also implemented by means of linear SVMs using the side-information vectors extracted from NIST'04 trials. The training procedure is carried out in both modes as described in Section III. We optimize $\{p_h, p_l, k\}$ for this evaluation set, then ultimately retrain the post-classifier using the optimized parameters and employ it to correct NIST'05 scores.

TABLE I
SYSTEMS PERFORMANCE

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
8	Ordinary Fusion SVM	4.29	93.96

A. Classifier Partitions

Our framework requires that the recognition systems used in fusion be split into two classes—roughly, low level and high level—that will form the basis for the post-classifier. As noted earlier, we performed a series of preliminary experiments in order to determine the optimal ways of splitting the several classifiers into either low- or high-level. It was observed that keeping System 1 as the sole low-level representative and assigning all others as high-level proved to be the best configuration, though only slightly better than also moving Systems 2 (and 3) to the low-level side.

The special role of the cepstral-GMM classifier relative to the other low-level classifiers may be rooted in two factors. First, the side-information vector used in these experiments is composed of statistics of cepstral parameters and thus directly connected to this feature set. On the other hand, this information is only indirectly related to the other low-level classifiers, since their feature sets are obtained through mathematical manipulations of the original cepstral parameters. Alternatively, it is probable that the best classifier split for this framework is simply to place the dominant classifier in the fusion scheme (System 1) in one partition and all others in the other partition.

Once we defined the low-level split consisting exclusively of System 1, we performed experiments in order to define an appropriate high-level partition composition. Specifically, we gradually dropped some of the other systems from the high-level partition, as shown in the next section.

B. "Oracle" Training

At this point, after we defined the low level-partition as containing solely System 1, we aim at investigating the effects of distinct classifier compositions in the high-level partition. Before performing our actual experiments, we would like to assess the *theoretical* upper boundaries of performance obtained through the distinct configurations, neutralizing the issue of $\{p_h, p_l, k\}$ optimization. We therefore optimized these parameters using the testing (and not the development) set for the distinct configurations. Then, for each configuration, the correspondent optimized parameters were used to train the post-classifier and correct the test-set scores. These "oracle" results are presented in Fig. 3 in terms of EER for both training modes and for the partitions evaluated. Some caution must be

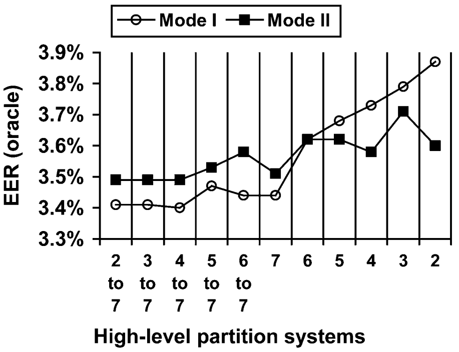


Fig. 3. "Oracle" performance for different high-level partitions.

TABLE II
OPTIMIZED TRAINING PARAMETERS

Training mode	$p_{h(\%)}$	$p_{l(\%)}$	k
I	15	35	0.60
II	5	10	0.45

exercised in the interpretation of results, since standard errors lie between 0.20% and 0.22% for each of these EER measurements (applying the binomial formula and error propagation for $\mathrm{EER} = (\mathrm{P_{fa}} + \mathrm{P_{miss}})/2$ as in [19]). In all cases, System 1 is the unique low-level representative. The high-level group might comprise all others systems ("2–7") or the most low-level candidates are gradually discarded ("3–7" and so on) until only individual systems are left.

The results suggest that Mode I exploits more efficiently the ensemble of high-level classifiers in order to improve the fused scores. On the other hand, training in Mode II seems to attain better performance when only single systems are available. In general, as expected, single high-level systems that are less correlated to the cepstral-GMM system (see Table I), although individually performing less well as speaker recognizers, perform better than low-level systems when used to train the post-classifier. In addition, the inclusion of other low-level systems to the high-level partition, as commented above, does not improve performance and they could eventually be discarded.

It is interesting to note that the set of optimized $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ is approximately constant for each of the configurations evaluated above and in particular for the best configurations. Typical optimized values for these configurations are depicted in Table II. Recall that there are ten times more impostor than target trials and therefore the relation between $\mathrm{p_h}$ and $\mathrm{p_l}$ values cannot be directly interpreted in terms of number of examples. The fact this set is stable for a variety of settings is particularly interesting for two reasons. First, this is evidence that the framework is robust to different systems and configurations. Second, this set could be used in future experiments as priors for actual optimizations, being helpful in avoiding convergence to local minima.

In order to assess the improvements obtained by the proposed framework in comparison to its basic version, we evaluated the performance of the basic ABIE system applied to the fused recognition scores. Note that the basic ABIE framework does not make any special assumptions to the fact that we are dealing with fused scores. On the other hand, ABIE adapted for fusion considers the relative performance of the constituent systems in order to correct the fused scores. The oracle performance for the basic ABIE framework was 3.70%, which compares unfavorably with the novel framework. Moreover, and more important, it was observed that this optimum performance was obtained for a relatively narrow range of $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ parameters. On the other hand, concerning the adapted framework, near optimum results were observed for a large range of $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ parameters. This is an additional indication of robustness in actual training conditions, which is analyzed in the next section.

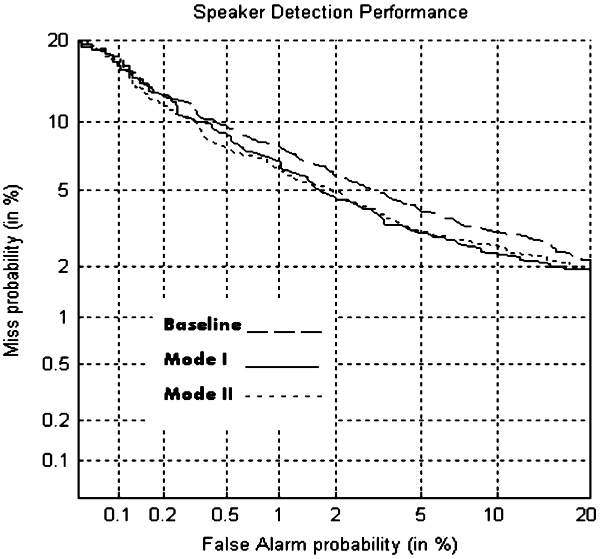


Fig. 4. Detection curve for ordinary and ABIE fusion.

C. Actual Training

In the previous section, we analyzed the best theoretical results our method can reach. This theoretically optimal result would be obtained if we could find the optimum set of parameters $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ for training the post-classifier given a specific low/high-level classifier partition. In fact, in real applications, we must find these parameters from a limited development set, and there are no guarantees we shall converge to the optimum set. We would currently like to investigate the abilities of the proposed parametric optimization method to match the oracle results.

We therefore reoptimize the post-classifiers, this time using the training set (NIST'04) and apply the optimized post-classifiers on the test set (NIST'05). We call this "actual" as opposed to the oracle optimization. We perform actual optimization via both the parametric and the original explicit method as explained in Section III. In either case, we end up with a series of triples $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ each one leading to a corresponding recognition error in the training set. We could expect that the triple attached to the minimum error will lead to the minimum error in the testing set as well. In practice, we noted that, occasionally, the optimization process leads to local minima. Fortunately, in general, spurious $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ candidates can often be detected

Mode I:

A. Channel mismatch at lower frequencies.

B. Quality mismatch at lower frequencies.

C. Average quality at higher frequencies.

D. Average quality at lower frequencies.

E. Pitch mismatch.

F. Speech rate mismatch.

G. Average speech rate.

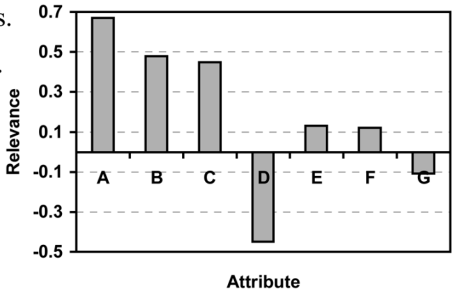


Fig. 5. Bias analysis for Mode I.

Mode II:

A. Quality mismatch at lower frequencies.

B. Channel mismatch at lower frequencies.

C. Average quality at lower frequencies.

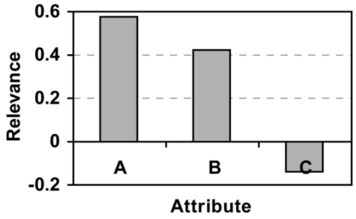


Fig. 6. Bias analysis for Mode II.

TABLE III
SUMMARY OF POST-CLASSIFICATION PERFORMANCE

Post-classifier	EER (%)	Improvement over baseline EER (%)	$\mathrm{C}_{\mathrm{det}}$ ($_{\mathrm{x}}10^4$)	Improvement over baseline Cdet (%)
None (baseline – ordinary fusion)	4.29	—	138	—
ABIE for fusion (Mode I)	3.41	20.52	131	5.07
ABIE for fusion (Mode II)	3.59	16.32	121	12.32

through visual inspection of the trends of the leading triples obtained for the training set.

As a rule, we noted that the parametric method surpasses its explicit version, obtaining results not far from the oracle. The best actual results were obtained for both training modes, leaving only System 1 in the low-level partition and assigning all other classifiers to the high-level partition (configuration "2–7" in Fig. 3). The detection curve for this configuration in both modes using parametric optimization, and for ordinary fusion (baseline) is depicted in Fig. 4.

The performance obtained by these three systems is summarized in Table III. Although the post-classifiers were optimized for minimum EER, we also present the minimum $\mathrm{C}_{\mathrm{Det}}$ (attained for the operating point leading to minimum EER). The $\mathrm{C}_{\mathrm{Det}}$ is a cost function defined by NIST, which typically focuses on the low False Alarm region performance [16]. In particular, Mode I, which aims at correcting the low-level classifiers' flaws attained the lowest EER and, in general, slightly improved performance on the Low Miss region. In contrast, Mode II, which emphasizes the high-level classifiers' hits attained better performance in the low False Alarm region, as confirmed by the $\mathrm{C}_{\mathrm{det}}$ parameter.

We further tried to combine both modes for training the positive trials. We believe the two training modes might be somewhat complementary, since they show uneven performance in different operating points. Specifically, we trained a single post-classifier assigning "$+1$" to both well classified high-level and poorly classified low-level trials and "-1" to well classified low-level and poorly classified high-level trials. The post-classifier's scores were then used to correct the fused scores. Unfortunately, this approach did not consistently perform as well as the two modes independently. Possibly, a simple "agglomeration" of the modes affects the linear separability of the combined regions, demanding more complex decision boundaries. In another attempt, simply weighting both scores did not simultaneously improve EER and $\mathrm{C}_{\mathrm{det}}$.

D. Bias Analysis

Bias analysis is a by-product of the presented framework. Once we have optimized and trained the post-classifier, an analysis of the post-classifier's weights can reveal which of the side-information elements is causing biased scores (see Section II-B). This is true for any classifier that allows some kind of inference regarding its decision rules. In particular, this is true for the SVM, which is used to implement the post-classifiers in these experiments. Figs. 5 and 6 show, for each training mode, the components of the side-information vector which obtained high magnitude weights in the SVM training. These attributes are presumably responsible for biased recognition scores. Since the side-information components are normalized to zero mean and unit standard deviation, the weights shown reflect the relative relevance of the distinct attributes in overall score bias.

The SVM output is formed by a weighted sum of the side-information vector components. Thus, highly positive weights lead to positive increments in the corrected scores [equations

(1a) and (1b)], once the correspondent vector components are high. On the other hand, highly negative weights lead to positive increments in the corrected scores, in case the correspondent vector components are also negative. The values depicted are the respective attribute weights normalized by the sum of weights attained by all side-information components.

It can be seen that the major bias sources are rooted in acoustic issues, particularly in train/test quality and channel mismatch in the lower-frequency portions of the spectrum. (For the current bias analysis, we roughly split the 19 Mel bands (Section II-B) into three regions: low, medium, and high.) In both cases, the post-classifier will boost scores whose trials suffer from high mismatch. (As noted above, although the post-classifier is trained exclusively on target trials, in practice, this correction can be similarly applied to impostor trials.) The same trend occurs with the average trial quality in lower frequency portions of the spectrum. Low-quality trials will also have their scores boosted for compensation.

To some extent, high-level attributes are also responsible for bias. Note that in Mode I the post-classifier is trained to boost scores whose trials show high mismatch in either pitch or speech rate. Since there are no cross-gender trials, pitch mismatch is related to changes in speaker mood or speaking style between train and test recordings. In addition, it can be observed that speakers who speak slowly will also have their scores boosted, since this seems to affect the low-level classifiers.

V. Virtual Fusion

In this section, we show that it is also possible to apply the proposed fusion post-classifier on scores of a single recognition system. In the training stage, we proceed exactly as above and use a variety of recognition systems for the ultimate purpose of learning optimal score corrections. However, in operating mode, we use only a single classifier (specifically, the low-level acoustic classifier), the scores of which are corrected based on the first stage. We call this "virtual fusion." The point is that extra classifiers are used offline only for training the post-classifier, so that one can explore the power of complex classification systems which might not be available at operation time. In operating mode, there is no need to obtain scores for these systems, but rather only the recognition scores of the specific classifier that are corrected by the post-classifier.

As before, we concentrate on the standard cepstral-GMM classifier (System 1 above) and use the other system as "auxiliary" classifiers in order to train the post-classifier. Hence, System 1 remains the sole representative of the low-level classifiers, and the other systems are assigned to the high-level partition. The optimization process remains the same as in Section III-B, except for the use of the cepstral-GMM scores and not the fused scores as the goal of optimization. In other words, S_t and S_i in equations (1a) and (1b) represent the target and impostor score distributions of System 1 and not of the fused scores as before.

We investigated the impact of a variety of high-level classifier partitions for training the post-classifier, similarly to that related in Section IV-B for the fused system. The results for both training modes are depicted in Fig. 7. For comparison, the oracle EER obtained by the basic ABIE approach is 6.38%. (Note that standard errors for EER measurements are estimated as above and lie between 0.27% and 0.28%, which are greater than some of the differences observed.)

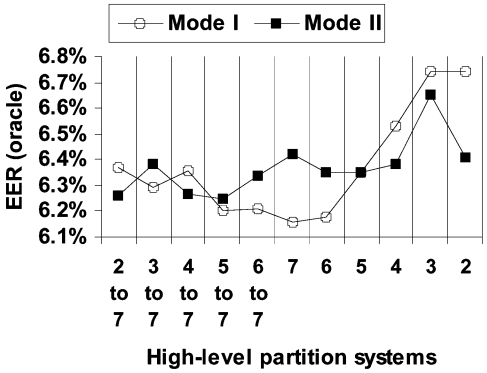


Fig. 7. "Oracle" performance for different high-level partitions.

Actual optimization was performed as well, and results were close to the oracle optimization. In particular, the best configurations: "5–7," "6–7," "6" and "7," trained in Mode I obtained actual EER in the range of 6.20% to 6.40%. The results suggest that appropriate auxiliary systems are those poorly correlated with the recognition system, whereas good performance seems to be quite irrelevant. Training the post-classifier using these systems slightly outperforms the basic ABIE method. Recall that in the basic ABIE, only held-out examples of System 1 are used to train the post-classifier, while in virtual fusion, examples of other classifiers are also used. This suggests that using side-information vectors from trials selected by higher-level systems enhances the post-classifier training. More significant, however, is the fact that optimum performance for the basic ABIE method is obtained for a relatively narrow range of $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ parameters, as noted in Section IV-B. On the other hand, in virtual fusion, near optimum results were observed for a large range of parameters. The optimum $\{\mathrm{p_h}, \mathrm{p_l}, \mathrm{k}\}$ values are relatively steady: $\{10\%, 30\%, 0.25\}$, which confirms the stability of this method. These values are similar to those obtained in Section IV-B for the fused scores, except for the constant k, which is weaker for the single system correction.

VI. Conclusion and Future Work

In this paper, we proposed a framework based on post-classification of scores intended for use with fusion of speaker recognition systems. In fact, we showed that this approach can be applied to single systems as well, provided we use a suitable "auxiliary" system in the training phase of the post-classifier ("virtual fusion"). The presented approach looks at speaker recognition fusion as a dual process, calibrating the fused score as a function of low- versus high-level classifier performance. In future work, we should look at ways to generalize this framework to handle more than these two classes.

At present, the post classifier can be trained in one of two training modes. In Mode I, it attempts to correct low-level classifiers flaws, and in Mode II, it enhances the high-level classifiers hits. We believe that merging both modes could further enhance performance.

The proposed framework was evaluated using NIST'04 and '05 speaker recognition benchmarks. Up to 20% decrease in EER was achieved compared to ordinary fusion of classifiers. It would be useful to corroborate these results by means of additional data-sets.

ACKNOWLEDGMENT

The authors would like to thank SRI International for kindly making their scores available. The authors also would like to thank the anonymous referees for numerous helpful comments on earlier drafts of this paper.

REFERENCES

[1] W. Andrews, M. A. Kohler, J. P. Campbell, and J. J. Godfrey, "Phonetic, idiolectal and acoustic speaker recognition," in *Proc. ODYSSEY-2001*, Crete, Greece, 2001, pp. 55–63.

[2] J. Campbell, D. Reynolds, and R. Dunn, "Fusing high- and low-level features for speaker recognition," in *Proc. 8th Eur. Conf. Speech Commun. Technol. (Eurospeech)*, Geneva, Switzerland, 2003, pp. 2665–2668.

[3] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang, "The SuperSID project: Exploiting high-level information for high-accuracy speaker recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Hong Kong, China, 2003, pp. 784–787.

[4] Y. Solewicz and M. Koppel, "Enhanced fusion methods for speaker verification," in *Proc. 9th Int. Conf. Speech Comput. (SPECOM'04)*, St. Petersburg, Russia, 2004, pp. 388–392.

[5] L. Ferrer, K. Sonmez, and S. Kajarekar, "Class-dependent score combination for speaker recognition," in *Proc. 9th Eur. Conf. Speech Commun. Technol. (Eurospeech)*, Lisbon, Portugal, 2005, pp. 2173–2176.

[6] Y. Solewicz and M. Koppel, "Considering speech quality in speaker verification fusion," in *Proc. 9th Eur. Conf. Speech Commun. Technol. (Eurospeech)*, Lisbon, Portugal, 2005, pp. 2189–2192.

[7] Y. Solewicz and M. Koppel, "Automatically correcting bias in speaker recognition systems," in *Proc. 16th IEEE Workshop Mach. Learn. Signal Process. (MLSP'2006)*, Maynooth, U.K., 2006, pp. 187–191.

[8] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, no. 2, pp. 254–272, Apr. 1981.

[9] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[10] O. Viikki and K. Lurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Commun.*, vol. 25, pp. 133–147, 1998.

[11] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. ODYSSEY-2001*, Crete, Greece, 2001, pp. 213–218.

[12] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Minneapolis, MN, 2003, pp. 53–56.

[13] F. Bimbot *et al.*, "A tutorial on text-independent speaker verification," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 4, pp. 430–451, 2004.

[14] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," in *Proc. Odyssey: Speaker Lang. Recognition Workshop*, Toledo, Spain, 2004, pp. 219–226.

[15] A. Solomonoff, C. Quillen, and W. M. Campbell, "Channel compensation for SVM speaker recognition," in *Proc. Odyssey: Speaker Lang. Recognition Workshop*, Toledo, Spain, 2004, pp. 57–62.

[16] NIST, "NIST- Speaker Recognition Evaluations." [Online]. Available: http://www.nist.gov/speech/tests/spk

[17] 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," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Toulouse, France, 2006, pp. 101–104.

[18] T. Joachims, , B. Schölkopf, C. Burges, and A. Smola, Eds., "Making large-scale SVM learning practical," in *Advances in Kernel Methods—Support Vector Learning*. Cambridge, MA: MIT Press, 1999.

[19] D. A. van Leeuwen, A. F. Martin, M. A. Przybocki, and J. S. Bouten, "NIST and NFI-TNO evaluation on automatic speaker recognition," *Comput. Speech Lang.*, vol. 20, pp. 128–158, 2005.

Yosef A. Solewicz received the B.Sc. degree in electrical engineering from The Technion–Israel Institute of Technology, Haifa, Israel, in 1990, the M.Sc. degree in electrical engineering from the Catholic University, Rio de Janeiro, Brazil, in 1993, and the Ph.D. degree in computer science from Bar-Ilan University, Ramat-Gan, Israel, in 2006.

Currently, he is a Postdoctoral Researcher at Bar-Ilan University and works with the Division of Identification and Forensic Science at the Israel National Police. His research interest is in machine learning with a focus on speaker recognition.

Moshe Koppel received the B.A. and M.A. degrees in mathematics from Yeshiva University, New York, NY, in 1977 and the Ph.D. degree in mathematics from New York University in 1979.

He was a Postdoctoral Fellow at the Institute for Advanced Study, Princeton, NJ, from 1979 to 1980 and has been on the Faculty of Mathematics and Computer Science at Bar-Ilan University, Ramat-Gan, Israel, since 1980. His current research focuses on text categorization, especially authorship attribution.

Prof. Koppel has served on the program committees of the IEEE Intelligence and Security Informatics Conference and numerous other conferences.