Class-dependent Score Combination for Speaker Recognition

Luciana Ferrer\textsuperscript{1} Kemal Sönmez\textsuperscript{2} Sachin Kajarekar\textsuperscript{2}

\textsuperscript{1} Department of Electrical Engineering
Stanford University, Stanford, CA, USA
\textsuperscript{2} Speech Technology and Research Laboratory
SRI International, Menlo Park, CA, USA
lferrer@stanford.edu \{kemal,sachin\}@speech.sri.com

Abstract
Many recent performance improvements in speaker recognition using higher-level features, as demonstrated in the NIST Speaker Recognition Evaluation (SRE) task, rely on combinations of multiple systems modeling a large variety of features. The diversity of the large set of features starting from short-term acoustic spectrum features all the way to habitual word usage from a large set of speakers in a multitude of settings (acoustic environment, speaking style, quantities of enrollment/test data) results in a challenging model combination task. In this work, we are presenting a class-dependent score combination technique that relies on clustering of both the target models and the test utterances in a vector space defined by a set of speaker-specific transformation parameters estimated during transcription of the talker’s speech by automatic speech recognition (ASR). We show that significant performance gains are obtained by using the first few principal components of a model transform for clustering the speaker verification trials into classes for (target speaker, test utterance) pairs, and then training a separate combiner for each class. We report results on the NIST SRE 2004 and FISHER datasets.

1. Introduction
Recent improvements in speaker recognition performance using higher-level features, as demonstrated in the NIST Speaker Recognition Evaluation (SRE) task, have come to rely on combinations of multiple speaker models of a variety of classes of features \cite{1}. The short-term acoustic spectrum features, the traditional basis of a GMM adapted speaker verification system, are combined at the score level with a large number of systems using features that cover various aspects of prosody and habitual word usage. The heterogeneity of the rich set of features used to model the large set of speakers and generate scores for speaker verification in a variety of settings (acoustic environment, speaking style, quantities of enrollment/test data) results in a challenging model combination task. The central goal of system combination is estimating a statistical mapping from the set of scores produced by individual systems to a combined score in order to reduce the equal error rate (EER) or the NIST-defined Detection Cost Function (DCF), the Bayesian risk with a specific set of parameters, both performance metrics summarizing the ultimate measure of performance, the ROC curve or the NIST-defined variation of the ROC curve, the Detection Error Tradeoff (DET) curve.

Many different techniques have been tried for score combination, the most common including linear combination, neural networks, perceptrons \cite{1,2}, and support vector machines \cite{3}. Each of these approaches estimates the parameters of a classifier for detecting true vs. impostor speech given the claimed target identity in the vector space of system scores being combined. The final score is either a (possibly non)linear regression on system scores or a distance from the separating boundary.

In designing a classifier in the score space, there is an alternative rationale based on the observation of distinct classes of target and test speakers or acoustical conditions, which motivates designing separate classifiers for each class. The class-dependent classifier scores are subsequently combined and/or selected to generate the final score. For example, in \cite{4}, trials are clustered according to the SNR of the speech waveform to train multiple classifiers during score combination of the cepstral and the higher-level systems.

In this work, we are presenting a class-dependent score combination technique that relies on clustering of both the target models and the test utterances in a vector space defined by a set of speaker-specific transformation parameters estimated during transcription of the talker’s speech by ASR. Specifically, we use the parameters of the maximum likelihood linear regression (MLLR) transform \cite{9}, which is estimated for each speaker during ASR. We map the components of the transform to a lower dimensional space via principal component analysis (PCA) and perform the clustering in this space. For each class in the product set, that is, (target, test) pair of clusters, we allocate a separate combiner trained to fit the data in the class. We show that significant performance gains are obtained by using the first few principal components of the MLLR transform for clustering all the speakers into a small number of classes for (target, test) pairs.

The paper is organized as follows. We describe the main infrastructure, i.e. the task, corpora, component systems, and the baseline combination setup in Section 2. The proposed method of class-dependent score combination is detailed in Section 3. Experiments on the NIST SRE task and analysis of the results are presented in Section 4, with a discussion concluding the paper in Section 5.

2. Baseline Setup

2.1. Task and corpora
The task on which we report performance of the class-dependent combination techniques is the speaker verification task as defined in the NIST SRE plan \cite{5}. We report results in the “common” condition, which has one conversation side (of a 5-minute conversation) as enrollment/training data for a target speaker, and one side as the testing utterance. The main criterion in the NIST SRE is the Detection Cost Function, defined as the Bayesian risk with $P_{\text{error}} = 0.01$, $C_{f}\|=1$, and $C_{m}=10$. This is the performance measure we use in this paper.
The FISHER development set is created from the FISHER database, which is collected and distributed by LDC for the DARPA EARS program. We selected two nonoverlapping sets of speakers from this data [5]. Each set is balanced with respect to different genders and handsets. The first set, comprising 1128 unique speakers, and a part of the SRE03 dataset involving 425 unique speakers were used to create the background models. The second FISHER set was divided into two equal splits, FISHER1 with 16579 trials and FISHER2 with 14599 trials, and used as train and development datasets. The 2004 NIST SRE dataset (referred to as SRE04) composed by 15318 trials is part of the conversational speech data recorded in the Mixer Project and was used for the final evaluation of the method. In the three databases, the proportion of true-speaker trials to impostor trials is 1/10.

2.2. ASR for speaker recognition features
The transcriptions and time alignments used for the long-term, higher-level features were generated with SRI's 5 times real time conversational telephone speech (CTS) recognition system, using improved models developed for the NIST RT-03F evaluation [6]. The word error rate (WER) on RT-03 evaluation data was 21%.

2.3. Component systems
Following is a list of the systems included in the combination with their performance on SRE04 (100*DCF / EER%).

**Cepstral GMM system.** (Perf: 3.37/8.01) This is a traditional background adapted cepstral GMM system that uses a 2048-component GMM and is described in detail in [6].

**Cepstral SVM system.** (Perf: 3.13/8.01) This system uses the baseline cepstral feature vector with CMS and transform-based channel normalization [7]. Four different systems modeling different projections of PCA-transformations of mean polynomial vectors (two of which are variance normalized) are combined with an equal weight to produce the final score.

**MLLR transform SVM system.** (Perf: 3.09/8.92) The MLLR-SVM system uses speaker adaptation transforms used in SRI's speech recognition system as features for speaker verification. The transform coefficients are modeled by SVMs [9].

**Phone n-gram system.** (Perf: 5.41/12.09) Phone bigram and trigram frequencies for each speaker are extracted from phone recognition lattices. The frequencies for the most frequent N-grams are combined into one feature vector and modeled with SVMs [8].

**Word N-gram SVM system.** (Perf: 8.06/23.05) This system uses a SVM with a linear kernel with word unigram, bigram, and trigram frequencies as features [6].

**SNERF system.** (Perf: 6.69/16.16) This system uses a set of prosodic features where the extraction region is defined by automatically estimated syllable boundaries. The modeling is done using SVMs [11].

**Duration system.** Three sets of duration features – state (Perf: 7.16/15.81), phone (Perf: 8.73/19.75), and word level (Perf: 8.62/21.50) – modeled by GMMs are used in this system [10].

2.4. Neural network combiner
The baseline combiner is a single-layer feed-forward network that uses a sigmoid output node during training and a linear output for the final predictions. The linear output allows better combination of these predictions with the ones from our class-dependent combiner. The scores from the component systems are normalized with respect to the statistics in the training set, and then used as inputs to the perceptron, which is trained by minimum squared error with output labels 0 (impostor) and 1 (target).

3. Class-dependent Score Combination

3.1. Main idea
The hypothesis that motivates the class-dependent approach is that there are distinct classes of pairs of background and training trials, and for each class, the combination of short-term spectral and long-term higher-level features may be optimized separately. We cluster target and testing conversation sides with identical schemes as described below, resulting in trial pairs specified by (target, test) clusters. Separate combiners are trained for each class, and during verification probabilities for each class are assigned to the trials and an appropriate mixture of combiners is used.

3.2. Features for clustering
For clustering the speakers we use as features the parameters of the two-class version of the MLLR transform as described in [9]. For each conversation side two affine transforms are estimated, one for the obstruents and one for the nonobstruents. Each transform contains a translation vector and a rotation matrix. These transforms can be viewed as a text-independent encapsulation of the speaker's acoustic properties. These values are concatenated to form a long vector of dimension 3120. Half of the background model data are then used to compute the principal components (PC's) of this vector. Finally, the vectors corresponding to the conversation sides on the other half of the background data and the ones used for train and test are transformed into the principal component space. Only the first few PC's are used during clustering.

Due to the speaker-dependent nature of the MLLR features, the obtained clusters contain speakers that are found by the recognizer to be acoustically similar. In particular, given that the MLLR transforms are computed in a gender-dependent manner by the recognizer, we expect this automatically determined gender to be strongly reflected in the value of these features. In fact, we observe that the first PC simply conveys the information of the gender as detected by the recognizer.

3.3. Clustering
We use the other half of the background data not used for the computation of the PC's to obtain the clusters by taking as feature vectors the first N PC's of the MLLR features and training a Gaussian mixture model (GMM) with diagonal covariance, equal volume, and equal shape for each Gaussian. The GMM determines the probability of each conversation side belonging to each of the clusters corresponding to the Gaussians.

The obtained clusters group the conversation sides into a small set of ”prototypical speakers”. We use these clusters to assign classes to each of the trials in the training and test databases. The class is determined by the cluster to which the target and the test conversation sides belong. For example, if the target conversation side has the highest probability of belonging to cluster 1 and the test side has the highest probability of belonging to cluster 2, then the class for that trial would be given by (1,2). Rather than making hard
3.4. Class-dependent combiners

Our goal is to train a model that is dependent on the class of the trial and generates a unique score for each trial given the nine-dimensional vector of individual scores. We train one combiner for each of the resulting classes (e.g., 4x4=16 of them if we are using four clusters), using all the samples to train all of the combiners, weighting the samples by the probability of belonging to each class times a factor that depends on the type of trial (true-speaker vs impostor) as explained below. This approach proved to be slightly more robust than using the training trials for which the probability for that class is the highest, especially when using more than two clusters. Linear combiners are trained to predict the labels of the trials (-1 for impostor and 1 for true-speaker) using weighted least squares (WLS) where the weight for each sample trial i when training the combiner corresponding to class j is given by

\[ w_{ij} = \frac{p_{ij} \cdot \text{prior}}{p_{ij} \cdot \text{prior} + s_{ij}} \]  

for true speaker trials

\[ w_{ij} = \frac{p_{ij} \cdot \text{prior}}{p_{ij} \cdot \text{prior} + s_{ij}} \]  

for impostor trials

where \( p_{ij} \) is the probability of trial i belonging to class j, \( \text{prior} \) is the true-speaker prior probability, \( s_{ij} \) is the impostor prior probability and \( s_{ij} \) and \( s_{ij} \) are the sum of the probabilities for all the impostor and true speaker trials for class j. Defining the weights in this way allows us to balance the data probability and for all the impostor and true speaker trials for class j.

In those classes. Hence, these classes correspond to the cases where the true-speaker trials were especially hard to the recognizer thought that the conversation sides were actually from the same speaker. In theory there should not be any "cross-cluster" trials because the task is designed to have only same-gender trials. However, since the genre as determined by the recognizer is not the same as the actual gender, there are enough trials in the cross-cluster classes to warrant separate linear classifiers.

Table 1 shows several statistics for the data in the classes and on the complete dataset. First and second rows show the number of impostor and true-speaker trials in each class. Note that the cross-cluster classes ((1,2) and (2,1)) have one order of magnitude less data than the same-cluster classes. As mentioned before, this is because there are no same-gender trials in the task. Only those trials for which the recognizer’s gender label does not match the true gender fall in those cases. Hence, these classes correspond to the cases where the true-speaker trials were especially hard to the degree that the recognizer thought that the conversation sides corresponded to two different genders, while they were actually from the same speaker.

Table 1: The breakdown of data and results for the class-independent and class-dependent combiners for Fisher data.

<table>
<thead>
<tr>
<th>Class</th>
<th>Number of Impostor Trials</th>
<th>Num. of True-Speaker Trials</th>
<th>EER for Class-Dep Classifier</th>
<th>EER for Class-Indep Classifier</th>
<th>Rel. Weight for Noncep Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>8456</td>
<td>794</td>
<td>1.97%</td>
<td>25.9%</td>
<td>8.1%</td>
</tr>
<tr>
<td>(1,2)</td>
<td>687</td>
<td>38</td>
<td>29.6%</td>
<td>0.65%</td>
<td>46.9%</td>
</tr>
<tr>
<td>(2,1)</td>
<td>402</td>
<td>38</td>
<td>22.6%</td>
<td>0.65%</td>
<td>14.8%</td>
</tr>
<tr>
<td>(2,2)</td>
<td>5525</td>
<td>638</td>
<td>0.65%</td>
<td>3.09%</td>
<td>42.0%</td>
</tr>
<tr>
<td>All</td>
<td>15070</td>
<td>1508</td>
<td>2.56%</td>
<td>3.09%</td>
<td>11.3%</td>
</tr>
</tbody>
</table>

Third and fourth rows show the EERs for the data in each of the classes using a class-independent classifier (trained using least squares regression on the complete data) and the class-dependent classifier, respectively. Analyzing the performance of each of the classifiers in the case of the class-independent combiner, we can observe that the cross-cluster cases are the hardest ones by far: with EER of more than 25% compared to less than 2% for the same-cluster cases. Note that it is for these trials that the class-dependent combiner makes the biggest difference, improving performance in those classes by up to 50% relative. Having a class-dependent combiner allows adaptation of the coefficients of the linear combination to the trials in the corresponding class. The last row in Table 1 shows the percent of the sum of coefficients assigned to the noncepstral systems (duration, snrf and
word-ngram) compared to the total sum of the coefficients for the classifiers trained using the weights for the corresponding class (in the case of all-data, the weights are one for all samples). Note that for the cross-cluster classes, the noncepstral features receive a much higher weight relative to the class-independent case. Finally, the performance in the same-cluster classes does not degrade with respect to that of the class-independent classifier.

Although the interpretation of what the clusters and the classes are is not as clear as for the two-cluster case, using three and four clusters slightly improves the performance. Table 2 shows the results for 4x4 and 2x2 clusters for comparison (performance with the 3x3 clusters is in between). In all cases, no more PC’s than the first three were found to be useful. The measure of performance we are optimizing is the DCF. In FISHER2, a relative improvement of 12% in DCF with respect to the neural network performance is achieved by averaging the neural network prediction with the class-dependent prediction. On SRE04, the relative improvement is around 7%. For this data, the number of false rejections (true-speaker trials misclassified as impostor trials) for the averaged combiner at the DCF point is significantly smaller (at 95% level) than for the neural network, while the number of false alarms is not significantly smaller (at 95% level) than for the neural network (true-speaker trials misclassified as impostor trials).

The relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%. For this data, the relative improvement is around 7%...