MLLR Transforms as Features in Speaker Recognition

Andreas Stolcke  Luciana Ferrer*  Sachin Kajarekar  Elizabeth Shriberg  Anand Venkataraman

Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, USA
*Department of Electrical Engineering, Stanford University, Stanford, CA, USA
{stolcke, lferrer, sachin, ees, anand}@speech.sri.com

1. Introduction

Current speaker recognition systems employ a combination of knowledge sources, but the basis of most state-of-the-art systems is still the modeling of cepstral features extracted over short time spans (a few tens of milliseconds) and modeled as an unordered set of independent samples. The modeling is typically carried out in terms of log-likelihood ratios of Gaussian mixtures [1], or discriminatively using support vector machines (SVMs) [2]. There are two fundamental problems with this approach. First, it ignores longer-term and higher-order structure in the speech, such as is best described at the level of phones, syllables, words, and whole utterances. Consequently, there have been numerous recent developments to characterize speaker idiosyncrasies at those levels, and state-of-the-art systems now typically employ a combination of long-term and short-term features [3, 4].

The second fundamental problem with short-term cepstral modeling is that the overall cepstral distribution confutes speaker characteristics with other factors, principally, channel properties and the choice of words spoken. Standard signal processing and feature-level normalization methods can alleviate some of the channel effects, and score-level normalization techniques such as HNORM [1] and TNORM [5] partially compensate for both sources of extraneous variability. Phone-conditioned (see [6] for an overview) and word-specific [7] cepstral models are a direct attempt to make models invariant to the choice of words (since words by and large determine the phone sequence). However, these approaches have the drawback of fragmenting the data and requiring sufficiently accurate speech recognition. Other recent work has also tried to explicitly decompose cepstral variability by source and design filters that are optimized for the factors that are desirable for a given task (e.g., speaker versus speech recognition) [8].

Although the speaker modeling approach proposed here is also based on cepstral features, it was motivated and enabled by our work on higher-level stylistic features, which typically require the use of large-vocabulary word recognition systems. Such systems use elaborate forms of adaptation to turn the speaker-independent recognition models into more accurate speaker-dependent models. Instead of modeling cepstral observations directly, we can model the “difference” between the speaker-dependent and the speaker-independent models. This difference is embodied by the coefficients of an affine transform of the Gaussian means in the recognition models. These transforms apply to models that are specific not only to phones, but to context-dependent phones (triphones). Thus, to the extent that the triphone-conditioned recognition models are independent of the choice of words, so are the speaker-specific transforms. Because the transforms themselves are shared among triphones (and to some extent also between phones), we avoid the problem of data fragmentation. We can thus represent the cepstral observations in a feature space of fixed, and relatively low, dimensionality. Furthermore, as we will show, the transform features lend themselves quite well to discriminative modeling with SVMs.

In the remainder of the paper we describe the details of our approach and explore several variants that arise in its implementation. We test the method on several speech databases, including the 2004 NIST speaker recognition evaluation (SRE) dataset, and compare its performance to that of standard cepstral models. Finally, we give results for combinations of the various models.

2. Method

2.1. Recognition system

Our speech recognition system is a fast, two-stage version of SRI’s conversational telephone speech (CTS) system, as originally developed for the 2003 DARPA Rich Transcription evaluation [9] and later modified for the NIST 2004 speaker recognition evaluation [3]. The system performs a first decoding using Mel frequency cepstral coefficient (MFCC) acoustic models and a bigram language model (LM), generating lattices which are then rescored with a higher-order LM. The resulting hypotheses are used to adapt a second set of models based on perceptual linear prediction (PLP) acoustic features. The adapted models are used in a second decoding pass that is constrained by trigram lattices, which generates N-best lists. These are then rescoring by a 4-gram LM and prosodic models to arrive at the final word hypotheses. The whole system runs in about 3 times real time on a hyperthreaded 3.4 GHz Intel Xeon processor.
2.2. Speaker adaptation transforms

In maximum likelihood linear regression (MLLR) [10], an affine transform \((A, b)\) is applied to the Gaussian mean vectors to map from speaker-independent \((\mu)\) to speaker-dependent \((\mu')\) means: \(\mu' = A\mu + b\). In unsupervised adaptation mode, the transform parameters (coefficients) are estimated so as to maximize the likelihood of the recognized speech under a preliminary recognition hypothesis. For a more detailed adaptation, the set of phone models can be partitioned or clustered by similarity, and a separate transform is applied to each cluster. In our system, MLLR is applied in both recognition passes. The first pass is based on a phone-loop model as reference, and uses three transforms, for nonspeech, obstructed, and nonobstructed phones, respectively. The second decoding pass uses a more detailed MLLR scheme, based on word references generated by the first pass, and nine different transforms corresponding to phone classes for nonspeech, voiced/unvoiced stops, voiced/unvoiced fricatives, high/low vowels, retroflex phones, and nasals.

2.3. Speaker-adaptive training

A variant of MLLR estimates transforms that apply to both Gaussian means and variances (constrained MLLR or CMLLR) [11]. The advantage of this approach that it is can be equivalently carried out by transforming the input features, rather than the model parameters. This makes it easier to apply the transforms on both training and test data, thus yielding models that normalize out training speaker variability, an approach known as feature-space MLLR (fMLLR) or speaker-adaptive training (SAT) [12]. The SRI system uses a single such transform (applied to all frames/phones) in the second decoding pass. This feature-space transform applies before the more detailed model-space transforms described above.

2.4. Feature extraction and SVM modeling

The coefficients from one or more adaptation transforms are concatenated into a single feature vector and modeled using support vector machines. The data used is from conversational telephone speech, and each conversation side is processed as a unit by the speech recognition system. Consequently, each conversation side produces a single set of adaptation transforms pertaining to the same speaker, and hence a single feature vector. Since our acoustic features (after dimensionality reduction) contain 39 components, the number of SVM feature components will equal the number of transforms \(39 \times 40\). In cases where the adaptation scheme uses a separate transform for non-speech models, that transform is left out of the feature vector, since it is not expected to help in speaker recognition.

An SVM is trained for each target speaker using the feature vectors from a background training set as negative examples (of which there are many, typically in the thousands), and the target speaker training data as positive examples (of which there are few, typically 1 or 8). To compensate for the severe imbalance between the target and background data, we adopted a cost model [13] to weight the positive examples 500-fold with respect to the negative examples. Throughout, a linear inner-product kernel function was used for SVM training.

We also found it advantageous to normalize the dynamic ranges of the feature vector components. This is necessary because the SVM kernel function is sensitive to the magnitude of the feature values, and hence to the relative weighting of feature dimensions. In the absence of prior information, a normalization procedure that roughly equates the dynamic ranges of feature components seems appropriate. We have had good success with two simple normalization methods. One is Z-normalization, which subtracts the means and divides by the standard deviations along each feature dimension. Another method is rank normalization, which replaces each feature value by its rank (normalized to the interval\([0, 1]\), i.e., the percentile) in the background distribution. Rank normalization performs an adaptive rescaling of the features to obtain an approximately uniform distribution. Rank normalization is computationally more expensive, but was found to work best in general; it was used in all reported experiments unless noted otherwise.

3. Experiments and Results

3.1. Datasets

We tested our baseline and MLLR-based systems on three databases: a subset of the NIST SRE-03 (Switchboard-II phase 2 and 3) data set, a selection of the Fisher collection conversations, and the NIST SRE-04 (Mixer) data. For Switchboard-II and SRE-04, two data sets were available, for training on 1 and 8 conversation sides, respectively. Table 1 summarizes the statistics of these data sets. The Switchboard-II trials were a subset of those used in the NIST SRE-03 evaluation, but had difficulty comparable to the full evaluation set, as measured by the performance of our baseline system.

The background training set consisted of 1553 conversation sides from Switchboard-II and Fisher that did not occur in (and did not share speakers with) any of the test sets, and that had duplicate speakers removed.

All data was processed identically by SRI’s speech recognition system as described above. None of the test or background data were used in training or tuning of the recognition system.

In addition to feature-level normalization, we performed TNORM score-level normalization [5] in all experiments, including for the baseline systems.

3.2. MLLR system results

We first tested systems based solely on the model adaptation transforms employed in the first and second recognition stages of our systems. The first stage uses two speech transforms, yielding a 3120-dimensional feature vector. The second stage
uses eight speech transforms, yielding a 12480-dimensional feature vector. We can also concatenate both these sets of transforms into a single 15600-dimensional feature. Table 2 summarizes the results in terms of both minimum detection cost function (DCF) and equal error rate (EER). DCF is the Bayesian risk function defined by NIST with $P_{\text{target}} = 0.1$, $C_{\text{fa}} = 1$, and $C_{\text{miss}} = 10$.

The results with 8-transform MLLR features are competitive with the best reported results for cepstral systems (cf. results in next section). Surprisingly good results are achieved by the 2-transform MLLR system, which uses only a simple phone-loop reference hypothesis, i.e., it does not rely on word recognition search. Finally, a consistent improvement over the 8-transform system is obtainable by concatenating the two feature vectors (using 10 transforms per speaker), showing that the two features are not entirely redundant. This may be in part because the two recognition stages (and corresponding MLLR) are based on different front-end features (MFCC versus PLP).

We also tried to optimize the number of transforms used in the second adaptation stage, since initially 8 just happened to be the value that was found to work best for speech recognition. However, no further improvement was obtained by either collapsing or refining the phone classes, indicating that the optimal choices for speech recognition and speaker recognition must be quite similar.

### 3.3. Baseline system combination

We compared the 10-transform MLLR system to two state-of-the-art cepstral systems. The first baseline system is a Gaussian mixture model (GMM) with universal background model (UBM) [1], based on 13 MFCCs (without C0) and first-, second-, and third-order difference features. The features are mean-subtracted and modeled by 2048 mixture components. Gender-handset models are adapted from this model and used for feature transformation [14]. The final features are mean and variance normalized at the utterance level. The detection score is the target/UBM likelihood ratio after TNORM.

The second baseline system is also based on MFCCs (with first- and second-order differences), followed by the same feature transformation and normalization steps. The final features are then modeled with SVMs utilizing the polynomial sequence kernel proposed by [2]. This baseline system shares the MLLR system the advantages of discriminative training and classification afforded by the SVM framework, but uses essentially the same features as the more traditional GMM-UBM system.

---

1. Note the cepstral SVM system used here is a standard one [2], not the enhanced version used by SRI in the 2005 NIST SRE.
Table 4: Speaker verification results on Fisher data, using SAT feature transforms.

<table>
<thead>
<tr>
<th>Features</th>
<th>Fisher 1-side</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-transform MLLR</td>
<td>5.50</td>
</tr>
<tr>
<td>after SAT norm</td>
<td>.08150</td>
</tr>
<tr>
<td>8-transform MLLR</td>
<td>5.64</td>
</tr>
<tr>
<td>without SAT norm</td>
<td>.08483</td>
</tr>
<tr>
<td>8-transform MLLR</td>
<td>5.50</td>
</tr>
<tr>
<td>+ 1-transform SAT</td>
<td>.08020</td>
</tr>
</tbody>
</table>

Table 5: Speaker verification results using MLLR and SAT transforms.

<table>
<thead>
<tr>
<th>Features</th>
<th>Fisher</th>
<th>SRE-04</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-side</td>
<td>1-side</td>
</tr>
<tr>
<td>MLLR (10 transforms)</td>
<td>5.50</td>
<td>8.92</td>
</tr>
<tr>
<td>MLLR + SAT (11 transforms)</td>
<td>.07818</td>
<td>.30949</td>
</tr>
<tr>
<td></td>
<td>5.50</td>
<td>9.07</td>
</tr>
<tr>
<td></td>
<td>.07686</td>
<td>.31287</td>
</tr>
</tbody>
</table>

We have yet to optimize the recognizer as a feature extractor for speaker recognition purposes. The present good results are achieved with features that are by-products of a system that was tuned for word recognition accuracy. It is quite possible that some of the other normalizations used (such as for vocal tract length) are in fact detrimental to speaker recognition.

5. Acknowledgments

We thank our colleagues Kemal Sonmez, for useful discussions, and Ramana Gadde, for clarifications of the MLLR implementation. This work was funded by a DoD KDD award via NSF IRI-9619921. The views herein are those of the authors and do not reflect the views of the funding agencies.

6. References


