

Speaker Recognition with Session Variability Normalization Based on MLLR Adaptation Transforms

Andreas Stolcke, *Senior Member, IEEE*, Sachin S. Kajarekar, Luciana Ferrer, Elizabeth Shriberg

Abstract—We present a new modeling approach for speaker recognition that uses the MLLR adaptation transforms employed by a speech recognition system as features for support vector machine (SVM) speaker models. This approach is attractive because, unlike standard frame-based cepstral speaker recognition models, it normalizes for the choice of spoken words in text-independent speaker verification without data fragmentation. We discuss the basics of the MLLR-SVM approach, and show how it can be enhanced by combining transforms relative to multiple reference models, with excellent results on recent English NIST evaluation sets. We then show how the approach can be applied even if no full word-level recognition system is available, which allows its use on non-English data even without matching speech recognizers. Finally, we examine how two recently proposed algorithms for intersession variability compensation perform in conjunction with MLLR-SVM.

Index Terms—Speaker recognition, MLLR-SVM, intersession variability compensation.

I. INTRODUCTION

THE dominant features used in current speaker recognition systems are 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 mixture models (GMMs) [1], or discriminatively using support vector machines (SVMs) [2]. A fundamental problem with short-term cepstral modeling is that the overall cepstral distribution conflates speaker characteristics with other factors, principally channel properties and the choice of words spoken. Signal processing and feature-level normalization methods, such as feature mapping [3], can alleviate some of the channel effects, and score-level normalization techniques such as HNORM [1] and TNORM [4] partially compensate for both sources of extraneous variability. Phone-conditioned (see [5] for an overview) and word-specific [6], [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 work has 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]. Recently, feature- and model-based compensation techniques for all types of intersession variability (which includes channel and text-dependency effects) have been proposed in the form of factor analysis for GMMs [9], and nuisance attribute projection (NAP) [10] and within-class covariance normalization (WCCN) [11] for SVMs.

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. Note that large-vocabulary conversational speech recognition systems have been proposed as components of speaker recognition systems predating the use of “higher-level” features and going back at least to [12]. 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.

We would also like to point out the conceptual similarity of MLLR-SVMs to another recent proposal for speaker modeling, based on Gaussian supervectors [13]. Both approaches define SVM features by characterizing cepstral observations in terms of a model adaptation process. The key differences are in the underlying speech model used (triphone hidden Markov models (HMMs) versus GMMs) and the adaptation method employed (MLLR versus MAP (maximum a posteriori) adaptation). We believe that the primary virtues of the MLLR-SVM are its use of a more detailed speech model and the efficient

Manuscript received February 15, 2007. This work was funded by a DoD KDD award via NSF IIS-0544682. The views herein are those of the authors and do not necessarily represent the views of the funding agency.

A. Stolcke, S. Kajarekar, L. Ferrer, and E. Shriberg are with the Speech Technology and Research Laboratory, SRI International, Menlo Park, CA 94025 USA. Corresponding author is A. Stolcke, phone: 510-666-2969; fax: 510-666-2956; e-mail: stolcke@speech.sri.com).

L. Ferrer is also with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

use of data through transforms that are shared among many Gaussians. The supervector SVM, on the other hand, does not require a speech recognizer and is therefore less complex to implement and language-independent.

At a more general level, MLLR-SVMs share similarities with other SVM approaches based on Fisher kernels [14] and probabilistic distance kernels [15]. Both of these approaches are based on defining generative sequence models that are then used to induce kernel functions for SVM modeling; as such, they are conceptually similar to our MLLR-based approach, which is also based on a generative model (HMMs) of the classified data.

In this paper we lay out the principles of MLLR-SVM speaker recognition, discuss its implementation in the context of SRI's speech and speaker recognition system, and give results on recent NIST speaker recognition evaluation data sets. We then show how our approach can be modified to require only a phone-based recognition model, making it applicable to languages for which no word-based recognition system is available.

Finally, we examine how intersession variability compensation with NAP and with WCCN compares when applied to MLLR features. We point out how both techniques may be viewed in a common framework, and explore a combination of both techniques. Results on NIST evaluation data show substantial gains on top of the basic MLLR-SVM approach.

II. MLLR-SVM SPEAKER RECOGNITION

A. Speaker verification framework

The specific speaker recognition task examined here is *speaker verification*, i.e., the detection of a known target speaker in a set of test samples, each of which comes either from the target speaker or from an unspecified (unlimited) set of impostor speakers. Each test sample is processed independently, without knowledge of the others. Following the NIST "extended data" SRE framework, both training and test samples consist of 5-minute-long recordings of one of the sides of a telephone conversation, thus providing about 2.5 minutes of speech on average [16]. Either 1 or 8 training conversation sides are provided for a given target speaker, implicitly defining a speaker *model*. For each model, several (target or impostor) test samples, each consisting of only one conversation side, are specified, are then given, each forming an independent *trial*. Results are reported as aggregates over a large number of models and trials.

The speaker verification paradigm we use is to define a discriminant function for each model. The value of the discriminant function on a test sample is positively correlated with the probability of the sample coming from the target speaker. Hence, the discriminant function value is compared to an empirically optimized threshold, and the sample is labeled a target sample if the threshold is exceeded.

The particular approach proposed here may be characterized by the following components: (a) the extraction of a (fixed-length) feature vector from a (variable-length) speech sample, which is accomplished via the speaker adaptation stage of speech recognizer, and (b) construction of a discriminant

function embodied by an SVM. Most of the complexity, and all of the novelty, of our approach is in the feature extraction, while the SVM modeling approach is well-known and widely used in conjunction with a range of speaker features [2], [17], [18], [13]. In the following subsections we describe the processing steps required for feature extraction and modeling in their temporal order.

B. Speech recognition system

Our speech recognition system is a simplified, two-stage version of SRI's conversational telephone speech (CTS) system, as originally developed for the 2003 DARPA Rich Transcription evaluation [19] and later modified for the NIST 2004 speaker recognition evaluation [18].¹ The system is trained on Switchboard 1, some Switchboard 2, and CallHome English data, as well as Broadcast News and web data for the language model.

A speech-nonspeech HMM is used to detect regions of speech, which form the basis of all processing, including that of the cepstral baseline speaker verification systems described later. The recognizer next adapts a set of within-word triphone models based on a 39-dimensional feature vectors computed from Mel frequency cepstral coefficients (MFCCs), first-, second-, and third-order differences, voicing features, and heteroscedastic linear discriminant analysis (HLDA) [20]. This step uses a phone-loop model for unsupervised MLLR speaker adaptation, and a bigram language model (LM) for decoding, generating lattices that are then rescored with a higher-order LM. The resulting hypotheses are used to adapt a second set of models, now consisting of cross-word triphones, based on perceptual linear prediction (PLP) acoustic features. The PLP front end uses first-, second-, and third-order differences, LDA, and a maximum likelihood linear transform (MLLT) [21] to yield another 39-dimensional feature vector. The second adaptation step uses more detailed transforms than the phone-loop adaptation step, as described in detail later. Both the MFCC and the PLP cepstrum are processed by vocal-tract-length normalization (VTLN) using the method of [22], and normalized for mean and variance at the conversation side level. In addition, the PLP models also use constrained MLLR [23] on training and test data (speaker-adaptive training, or SAT) [24] for feature-level normalization. While VTLN and SAT could potentially detract from speaker discrimination via MLLR transforms as it renders different speakers more alike, we initially left these components of the recognition system unchanged for expediency, and later verified that they did not, in fact, degrade speaker recognition performance.

The recognition system then performs additional decoding and rescoring passes to arrive at its final word hypotheses, but these steps are irrelevant for the MLLR-SVM system since only the hypotheses from the first pass are used for MLLR. This system has a final word error rate (WER) of about 21% on Fisher CTS data; the MLLR adaptation hypotheses have a

¹The system is also somewhat outdated as it does not make use of all the latest available training data and modeling techniques, but was kept unchanged from prior years for expediency.

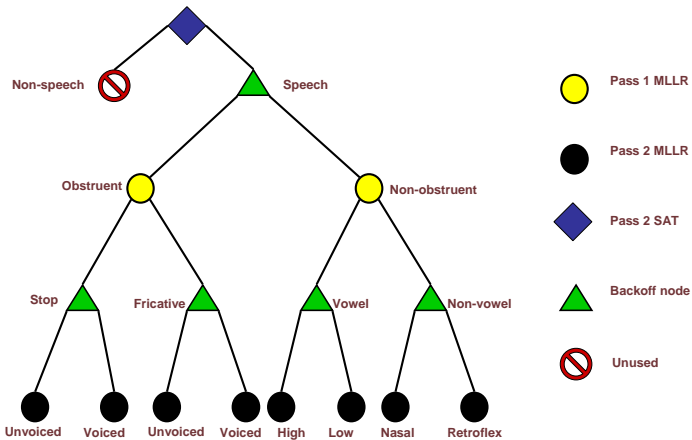


Fig. 1. Regression class tree used in MLLR.

WER of about 29%.² The whole system runs in about 3 times real time on 3.4 GHz Intel Xeon processor.

C. Speaker adaptation transforms

In maximum likelihood linear regression (MLLR) [25], [26], an affine transform (A, b) is applied to the Gaussian mean vectors to map from speaker-independent (μ) to speaker-dependent (μ') means:

$$\mu' = A\mu + b$$

where A is a full matrix and b a vector. 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. The parameters A and b are shared among all speech models, making the estimation robust to small amounts of adaptation data. 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, obstruent, and nonobstruent 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. Figure 1 depicts the regression class tree used by the two MLLR stages, with each tree node corresponding to a phone class with associated transform. If not enough adaptation data (200 frames) is available to a certain class for a given speaker, the MLLR algorithm backs off to the nearest parent class that does have the required amount of data and uses the transform for that parent class.

It should be noted that none of the parameters of the recognition system, including the number of MLLR phone

²The error rate on SRE data is not known since that data is not transcribed; however it is expected to be significantly higher on recent SRE (Mixer) data due to the temporal distance to the training data, as well as the more varied nature of the speaker population.

classes and the minimum number of frames, where optimized in the context of the speaker verification system. Rather, these parameters had been previously optimized for word recognition accuracy, on independent test sets. This was done for expediency and to avoid having to run two separate recognition systems for MLLR-SVM and other speaker recognition models [27]. However, we did make a post-hoc attempt to improve MLLR-SVM performance by defining alternate MLLR regression classes and varying the number of classes. Significantly, we were not able to achieve improvements over the originally defined 8 regression classes depicted in Figure 1.

Also, as described in Section II-B, in the second recognition pass, a single feature-level transform is used to effect speaker-adaptive training (SAT) [24]. In preliminary work [28] we found that these transforms are helpful in normalizing out corpus and channel differences, and lead to suboptimal results when used for speaker modeling in the MLLR-SVM framework.

D. 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 with HLDA or LDA/MLLT) contain 39 components, the number of SVM feature components will equal the number of transforms $\times 39 \times 40$. The transform for nonspeech (pause) models is left out of the feature vector, since it cannot be expected to help in speaker recognition. Note that due to the backing-off scheme used in transform estimation, some of the component values of the final feature vector might be tied (i.e., duplicated) for speakers with small amounts of data.

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, 1 or 8 in our framework). Throughout, a linear inner-product kernel function was used for SVM training. Our implementation is based on a modified version of the SVM^{light} software [29].

E. Feature normalization

Prior to SVM training or testing, features need to be normalized to equate their dynamic ranges. To this end, we apply rank normalization, replacing each feature value by its rank among all the background data samples on a given dimension, and then scaling ranks to a value between 0 and 1. Rank normalization not only scales the feature distribution to a fixed interval, it also warps the distribution to be approximately uniform. This has the intuitive effect that the distance between two datapoints (along a single dimension) becomes proportional to the percentage of the population that lies between them.

An alternative to feature normalization is to optimize the kernel function explicitly for minimal classification error. This can be done, for example, by applying scaling factors to subfeature-vectors [30]; however, we have not used this technique here since it is computationally expensive. The WCCN intersession variability compensation method investigated in Section III can be derived as an optimization of the SVM kernel to minimize an upper bound on classification error [11]. In practice it seems to work best to apply these kernel optimization methods in addition to (i.e., after) rank normalization has been applied.

F. Multiple MLLR reference models

MLLR transforms are computed relative to a reference model representing the “average” speaker. An ASR system might use different recognition models for different speakers, e.g., depending on gender differences and domain differences (noise conditions, genre of speech, etc.). For speaker recognition purposes it is important to compute all MLLR transform features using the same reference model to ensure comparability. However, the availability of different recognition models raises the possibility of expanding the feature space by computing MLLR transforms relative to an array of reference models and concatenating the resulting feature vectors into a “supervector”.

In general the different recognition models are not just linear transforms of each other. Therefore we can expect the corresponding sets of MLLR features to afford different, not entirely redundant “views” of the observation space, and the resulting combined feature vector to yield higher accuracy.

We explored this option by processing all speakers with both male and female gender-dependent recognition models used in our recognition system. The recognition hypotheses were always computed using the most likely gender assignment, but gender-dependent adaptation steps (including vocal-tract length normalization and MLLR) were computed separately for each gender. We informally refer to the resulting features as “male” and “female” transforms.

G. Baseline systems

In evaluating MLLR-feature-based speaker recognition systems, we compared results to two state-of-the-art cepstral feature 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 [3]. The final features are mean and variance normalized at the utterance level. The detection score is the target/UBM likelihood ratio after TNORM [4].

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], with some recently developed enhancements [31]. Principal component analysis is

performed on the polynomial feature space, and the features are projected onto the subspace spanned by the background speaker set, as well as its orthogonal complement (there are more feature dimensions than background speakers). This process is then carried out twice, for two different feature normalization variants, and four separate SVM models are trained. The overall system score is the sum of the four SVM scores, after TNORM.

H. Datasets

We tested our baseline and MLLR-based systems on four databases: a subset of the NIST SRE-03 (Switchboard-II phase 2 and 3) data set, a selection of Fisher collection conversations, and the NIST SRE-04, SRE-05 and SRE-06 data sets. For all but Fisher, two data sets were available, for training on 1 and 8 conversation sides, respectively. The NIST SRE-04, SRE-05, and SRE-06 data sets were drawn from the Mixer data collection [16], which included telephone conversations in English as well as other languages. Since our method relies on a speech recognizer for English, we report on trials that involve only English conversations. For SRE-05 and SRE-06 we chose the primary evaluation (Common Condition) subset, which is likewise English-only. Table I summarizes the statistics of these data sets. Note that 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.

For all SRE sets, the evaluation is performed with 2.5 minutes of training data and another 2.5 minutes of testing data, on average, since each data point is obtained from one of the sides of a 5-minute-long conversation. However, Fisher conversations used as part of the background set were about 10 minutes long, and therefore had about 5 minutes of speech per speaker.

All data was processed identically by SRI’s speech recognition system as described in Section II-B. Note that none of the test or background data had been used in training or tuning of the recognition system.

In addition to feature-level normalization, we performed TNORM score-level normalization [4] in all experiments and for all systems, using speaker models drawn from a separate Fisher data set.

Systems were optimized using the Fisher and SRE-04 data sets, and we give results on these to illustrate certain contrasts that guided our development. Final systems are then tested on all data sets.

I. Results

Table II summarizes development results for the 8 MLLR transforms obtained in the second recognition pass, in terms of both minimum detection cost function (DCF)³ and equal error rate (EER).

³DCF is the Bayes risk function defined by NIST with $P_{\text{target}} = 0.1$, $C_{\text{fa}} = 1$, and $C_{\text{miss}} = 10$.

TABLE I
DATA SETS USED IN EXPERIMENTS

Test set	SWB-II		Fisher	SRE-04 English-only		SRE-05 Common Condition		SRE-06 Common Condition	
Training	1-side	8-side	1-side	1-side	8-side	1-side	8-side	1-side	8-side
Conv. sides	3642	3058	734	1384	2695	2628	5205	3209	6556
Models	578	546	734	479	225	506	384	517	483
Trials	9765	4911	16578	15317	7336	20907	15947	24013	17547

TABLE II
SPEAKER VERIFICATION RESULTS USING MLLR FEATURES FROM SECOND ADAPTATION STAGE (8 TRANSFORMS). THE TOP NUMBER (IN ITALICS>) IN EACH TABLE CELL IS THE EER (%). THE BOTTOM NUMBER IS THE MINIMUM DCF VALUE.

	Fisher	SRE-04	
MLLR gender	1-side	1-side	8-side
Male	2.92 .06095	6.25 .28812	3.21 .12053
Female	2.98 .05362	6.54 .29092	3.21 .14568
Male + Female	2.85 .05493	5.34 .25640	2.62 .11767

The three rows of results correspond to MLLR-SVM systems using male, female, and concatenated male+female transforms, respectively. The feature dimensionality was $39 \times 40 \times 8 = 12480$ for the single-gender systems, and 24960 for the combined system. Male and female transforms give approximately equal results. Combining the two sets of transforms yields a substantial gain: EERs on SRE-04 are reduced by an additional 12% to 21% relative, and DCFs by 2% to 20%.

Next we examine results in comparison to our baselines. The top part of Table III gives complete results for the cepstral GMM and SVM systems, as well as for the MLLR system using 8+8 (male+female) transforms. We observe that the results across all data sets are quite consistent, and, in particular, SRE-05 results are very similar to those on SRE-04. The cepstral GMM is competitive with the cepstral SVM in the 1-side training condition, but falls significantly behind the two SVM systems in the 8-side condition.

The bottom part of Table III shows results with combinations of the two MFCC baseline systems with the MLLR-SVM systems, using a neural network for combining the system output scores. The combiner is trained to minimize DCF on the SRE-04 data sets. Figures 2 and 3 plot the detection error tradeoffs for the two baselines, the 8+8 MLLR-SVM system, as well as for the three-way combined system.

As shown in the bottom part of the table, combining the 8+8-transform MLLR-SVM system with one of the cepstral systems generally yields sizable improvements over the MLLR-SVM system by itself. By contrast, a combination of the two baseline systems yields a much smaller error reduction over the individual baselines, showing that system combination *per se* is not sufficient to obtain good results, and that the MLLR-SVM system contributes information that complements the baselines. A three-way combination does, however, improve over the best two-way system in DCF (for which the combiner was optimized). The two-way combination of cepstral SVM and MLLR-SVM had the best EER, yielding

24% (for 1-side training) and 12% (for 8-side training) relative EER reduction over the MLLR-SVM system by itself.

J. MLLR-SVM without word recognition

We investigated using the less-detailed MLLR transforms from the first recognition stage of our ASR system. Since no prior recognition hypotheses are available at this stage, the MLLR algorithm maximizes the data likelihood using a loop over all phones, with context-independent (monophone) acoustic models. This also makes for a less-detailed modeling of the speech, but has the advantage of simplicity and speed. As with the stage-2, 8-transform MLLR-SVM system, we concatenated male and female transforms for best results.

Table III shows results of the GMM baseline, the 2+2 MLLR-SVM system, and a neural-network combination of the two. Interestingly, the 2+2-transform MLLR-SVM system is competitive with the MFCC GMM system, and beats it in the 8-side condition. The bottom row in Table III shows that even the 2+2-transform MLLR-SVM systems can boost the accuracy of a GMM baseline system significantly when combined with the latter. This might be of interest if full word recognition is not an option, as transforms here are computed using only a simple phone-loop decoding pass.

We also tried combining the 2+2 MLLR-SVM system with the 8+8 system, but found no additional gains. Consequently, the step-2 MLLR-SVM system with 8+8 transforms is the system of choice for now, assuming a full recognizer can be run. The step-1 MLLR system is still of interest, for example, when a full recognizer is too costly to run, or for mixed-language speaker verification.

To validate the phone-loop MLLR-SVM system for non-English use, we applied it to an Arabic speaker recognition task. A test set was constructed from the Arabic-language conversations contained in the NIST SRE-04 and SRE-05 (Mixer) evaluation sets, comprising 594 target trials and 5940 impostor trials. For background training data we used a collection of 1153 conversation sides compiled from the LDC's CallHome Egyptian Colloquial Arabic and Levantine Conversational Arabic corpora, as well as otherwise unused Arabic Mixer conversations. The phone-loop MLLR-SVM system (with combined male and female transforms) obtained an EER of 8.41% on this test set, which compared favorably to an EER of 9.09% for a cepstral GMM system trained on the same data. Note that the MLLR-SVM was based on English phone models, whereas the cepstral GMM has no language dependencies.

TABLE III

SPEAKER VERIFICATION RESULTS USING BASELINE, MLLR-SVM, AND COMBINED SYSTEMS. THE TOP VALUE IN EACH CELL IS THE EER, BELOW IT THE MINIMUM DCF VALUE APPEARS IN NORMAL FONT. FOR SRE-05, THE ACTUAL DCF VALUES USING THRESHOLDS OPTIMIZED ON SRE-04 ARE SHOWN IN **boldface**. THE MLLR-SVM SYSTEM USES 8+8 TRANSFORMS (SAME AS LAST ROW IN TABLE II). THE LAST ROW REPRESENTS A THREE-WAY COMBINED SYSTEM.

System	SWB-II		Fisher	SRE-04		SRE-05	
	1-side	8-side	1-side	1-side	8-side	1-side	8-side
MFCC GMM	4.63 .17857	1.92 .08353	4.57 .10259	7.77 .31126	4.95 .21146	7.22 .24775 .25807	4.97 .16876 .18386
MFCC SVM	4.38 .15610	1.06 .04470	4.31 .11051	8.01 .31339	3.33 .12629	7.26 .26833 .28433	3.11 .10324 .11853
MLLR (8+8) SVM	3.00 .10759	0.48 .02419	2.85 .05493	5.34 .25640	2.62 .11767	5.91 .17943 .21365	2.45 .07908 .09166
MFCC GMM + MFCC SVM						5.77 .21479 .22049	3.17 .10270 .11503
MFCC GMM +MLLR (8+8) SVM						4.84 .15202 .16608	2.45 .07086 .09494
MFCC SVM +MLLR (8+8) SVM						4.52 .15774 .19177	2.27 .06323 .09122
MFCC (GMM+SVM) +MLLR (8+8) SVM						4.61 .15038 .17188	2.21 .06332 .09122

TABLE IV

SPEAKER VERIFICATION RESULTS USING BASELINE SYSTEM AND MLLR-SVM BASED ON 2+2 TRANSFORMS FROM FIRST RECOGNITION PASS. THE TOP VALUE IN EACH CELL IS THE EER, BELOW IT THE MINIMUM DCF VALUE APPEARS IN NORMAL FONT. FOR SRE-05, THE ACTUAL DCF VALUES USING THRESHOLDS OPTIMIZED ON SRE-04 ARE SHOWN IN **boldface**.

System	SWB-II		Fisher	SRE-04		SRE-05	
	1-side	8-side	1-side	1-side	8-side	1-side	8-side
MFCC GMM	4.63 .17857	1.92 .08353	4.57 .10259	7.77 .31126	4.95 .21146	7.22 .24775 .25807	4.97 .16876 .18386
MLLR (2+2) SVM	4.72 .18130	1.12 .06387	2.95 .05756	8.22 .33962	4.37 .16283	7.96 .29527 .31496	4.55 .15737 .16622
MFCC GMM +MLLR (2+2) SVM						6.38 .23205 .24491	4.01 .13176 .14937

III. SESSION VARIABILITY NORMALIZATION

A. Background

Kenny [9], [32] showed that intersession variability (ISV) is an important source of mismatch in speaker recognition. ISV is the (averaged) variation between different conversations by the same speaker, and thus subsumes variation due to channel, speaker (e.g., emotional) state, speaking style, and phonetic content. Kenny proposed a factor analysis model, and other researchers have confirmed its effectiveness for GMM-based systems [33].

In parallel, researchers have developed methods for ISV modeling within the SVM framework. NAP was originally shown effective on cepstral SVM systems [34], [10] whereas WCCN was validated on a version of our MLLR-SVM system [11]. WCCN is theoretically more sound, but also computationally more cumbersome. Here, we aim to give a theoretical and empirical side-by-side comparison of the two techniques.

It is important to note some key properties of the eigenanalysis performed for both WCCN and NAP. The dimensionality

of MLLR features (T) is around 20k, and the number of conversations (M) available for computing the within-speaker covariance matrix is around 3k to 6k. It is computationally impossible to perform an eigenanalysis of a $T \times T$ covariance matrix, which itself is an ill-conditioned matrix and has only $M - 1$ nonzero eigenvalues. Therefore, a kernel trick is used [35]. The eigenanalysis is performed with an $M \times M$ covariance matrix in the conversation space, and the eigenvectors are transformed back to the original feature space.

B. Within-class covariance normalization

It is convenient to describe WCCN first and subsequently interpret NAP as a simplified WCCN. Hatch et al. [11] proposed the WCCN approach and showed significant improvements on SRE-04 and SRE-05 data. Figure 4 depicts how the ISV is computed (“Analysis”) and how the result is used for normalization (“Application”). Note that this is only a brief overview of WCCN; for more a detailed explanation and derivation see [11].

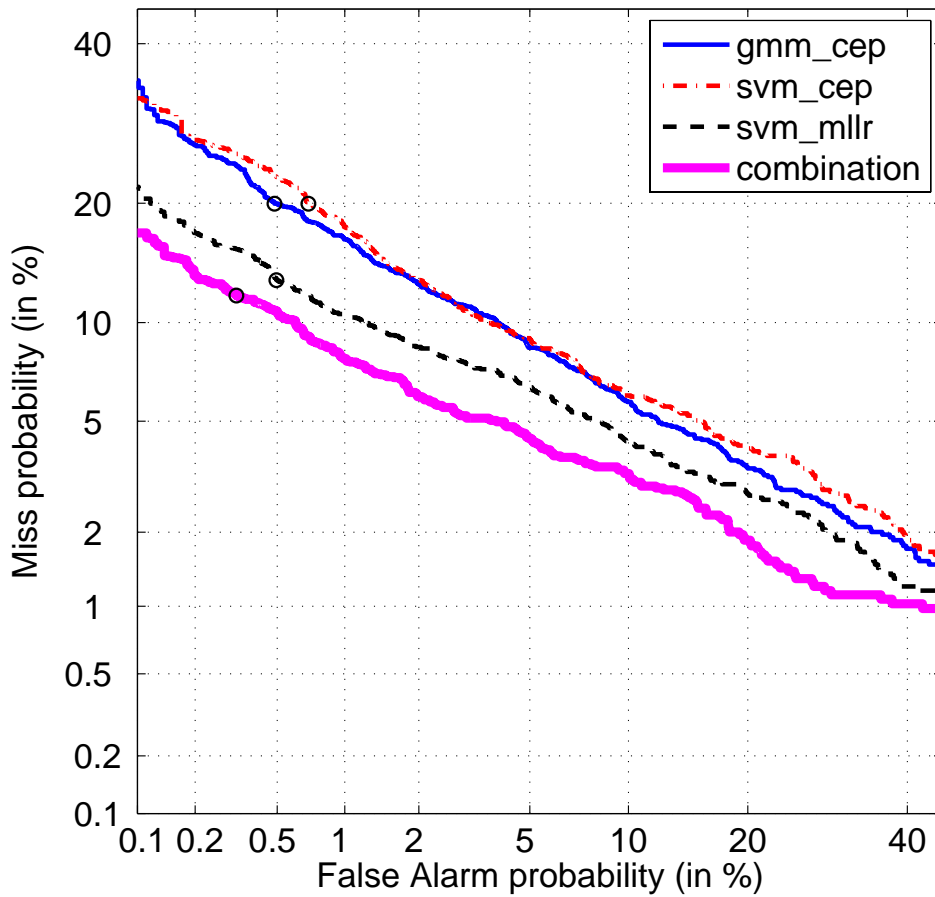


Fig. 2. Detection error tradeoff (DET) curves for baseline, 8+8-transform MLLR, and combined systems, SRE-05 1-conversation-side condition.

In the analysis part, the MLLR features on the ISV data set are normalized by within-speaker variance (WSV), to ensure the proper conditioning of the within-speaker covariance matrix (WSCM) estimated in the next step. Eigenanalysis is performed on this covariance matrix and a set of eigenvectors (E_M) is computed using the kernel method. WSV-normalized features are projected onto these eigenvectors and the within-speaker variance is again computed in the transformed space (WSV_1).

During the application, the features are first normalized with WSV and projected onto E_M . The projections (V_1) have dimensionality M . Using the projections, a complement feature vector (V_2) is created by subtracting the reconstructed feature vector in E_M space from the original feature vector. This vector has dimensionality T . V_1 is normalized by a function of WSV_1 ,

$$f(WSV_1) = \alpha - (1 - \alpha)\sqrt{WSV_1}$$

and V_2 is weighted by a scalar β . Finally, the weighted V_1 and V_2 are concatenated to form a single $M + T$ -dimensional feature vector. The scalars α and β are chosen on a development set.

C. Nuisance attribute projection

NAP as proposed by Solomonoff et al. [34] is based on principal component analysis and local linear embedding. The assumption is that unwanted variability can be sufficiently estimated in a high-dimensional feature space using second order statistics (the covariance matrix). Further, it is assumed that this variability lies in a lower-dimensional subspace spanned by the eigenvectors of the covariance matrix. Thus, one way to suppress the variability is to estimate this lower-dimensional subspace and remove it. Solomonoff et al. [10] have studied this approach extensively with many different ways of obtaining the covariance matrix. Recently, Burget et al. [36] adopted NAP and applied it to SVMs based on GMM-supervector and MLLR features.

Figure 5 shows the analysis and application of NAP with a WCCN-like template to highlight the similarities between the two approaches. The analysis involves simply computing the WSCM and computing its eigenvectors. As mentioned earlier, the first N eigenvectors are ignored and the feature vector is reconstructed in the original feature space. N is chosen so as to optimize performance on the development set.

D. Comparison of WCCN and NAP

Apart from the WSV normalization for WCCN, the two techniques differ mainly in how the different eigenvectors are

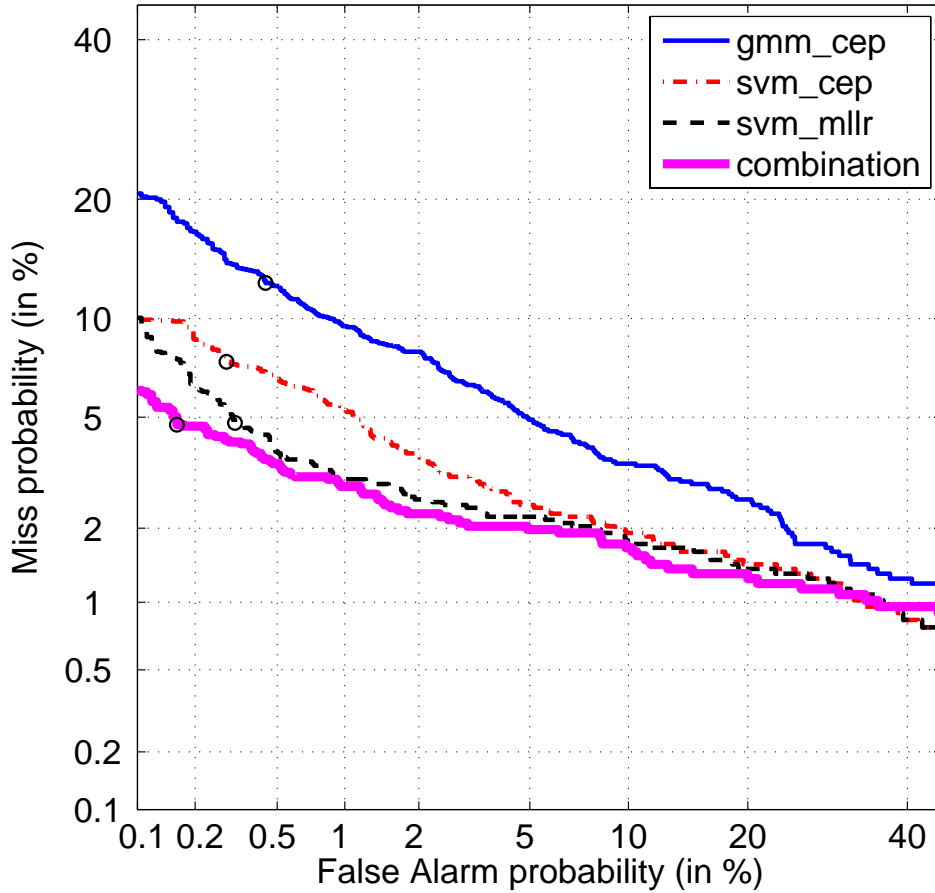


Fig. 3. Detection error tradeoff (DET) curves for baseline, 8+8-transform MLLR, and combined systems, SRE-05 8-conversation side condition.

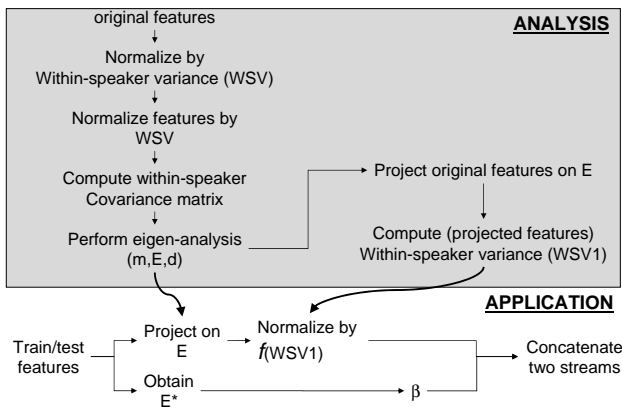


Fig. 4. Block diagram for WCCN processing.

weighted. Since the total number of eigenvectors is T , we can partition the eigenbasis E_T such that

$$E = E_T = [E_M, E_{M \rightarrow T}]$$

where E_M are the leading nonzero eigenvectors and $E_{M \rightarrow T}$ are the eigenvectors corresponding to the zero eigenvalues. For practical purposes, $E_{M \rightarrow T}$ are not computed explicitly but as

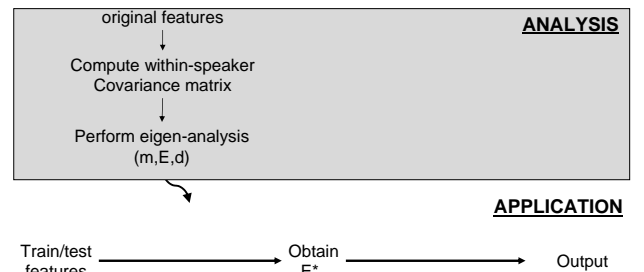


Fig. 5. Block diagram for NAP processing.

the complement of E_M :

$$xE_{M \rightarrow T} = x(I - E_M E_M^t) = xE_M^*$$

where x is the MLLR feature vector.

A generic framework for generating new features can be defined as

$$x' = [f_M x E_M, C x E_M^*]$$

where f_M is a function that generates weightings for M eigenvectors and C is a constant used to weight the complement. With $f_M = f_{WSV}$, the features correspond to WCCN features. With $M = N$, $f_N = 0$ and $C = 1$, we obtain NAP

TABLE V

WCCN RESULTS (N/A INDICATES BASELINE RESULTS WITHOUT WCCN). NUMBERS IN BOLD SHOW THE BEST PERFORMANCE FOR EACH TEST SET. NUMBERS IN PARENTHESES IN THE LAST ROW SHOW RESULTS IF PARAMETERS ARE TUNED ON SRE-06 INSTEAD OF ON SRE-05.

Background data	ISV data	SRE-05	SRE-06
		1-side	1-side
SWB+Fisher	N/A	5.872 .190	4.639 .224
	SRE-03	5.066 .154	4.314 .198
	SRE-04	5.056 .147	4.477 .216
SRE-04	N/A	6.189 .200	4.315 .197
	SRE-03	5.219 .162	3.776 .173
	SRE-04	5.103 .157	3.603 (3.452) .166 (0.162)

features. With $f_M = C_1$ and $C = C_2$, as two constants, we obtain features proposed by [31], which we will revisit in Section III-G.

E. Data issues

To estimate intersession variability a corpus of speakers with multiple conversations (sessions) per speaker is required. Thus, we selected speakers with at least eight conversations as our “ISV data set”. SRE-03 (Switchboard-II) data has about 625 such speakers, and SRE-04 (Mixer) data has about 310 such speakers. We used two sets of negative training samples (background speakers) for SVM training, so as to be compatible with previous results: the Switchboard+Fisher set used so far and described in Section II-H, and a new background set comprising all English SRE-04 speakers. This also allowed us to investigate the effect of mismatch between background and test data. Data from SRE-05 and SRE-06 were reserved for evaluation, with the former used to tune the parameters for each method, and the latter used to test generalization.

F. Results

We estimate WCCN parameters (α and β) and NAP parameters (N) on SRE-05 and apply the parameters to SRE-06. To reduce computational demands and keep the analyses from becoming very complex, results in this section do not make use of TNORM score normalization, and are given for the 1-side training condition only. We found that TNORM has only minimal or no benefit for the MLLR-SVM after ISV compensation.⁴

Table V shows the WCCN results for different conditions. Note that SRE-05 shows optimal results using this technique since it was used to optimize the free parameters, while SRE-06 results are an indication of how the technique generalizes to new data. We observe that WCCN consistently gives 13% to 17% improvement over the baseline (row N/A) on SRE-05 data. The best performance is obtained using Switchboard

⁴TNORM would also add a third dimension to the choice of training data, in addition to those corresponding to the choice of background and ISV data.

TABLE VI

WCCN RESULTS WITH SESSION VARIABILITY COMPUTED ON SRE-04 DATA SHOWING BREAKDOWN OF RESULTS ($E = E_M$).

Background data	ISV compensation	SRE-05	SRE-06
		1-side	1-side
SWB+Fisher	None	5.872 .190	4.639 .225
	$E_{xf}(WSV_1)$	5.542 .182	5.556 .266
	$+E^*x\beta$	5.056 .147	4.477 .216
SRE-04	None	6.189 .200	4.315 .197
	$E_{xf}(WSV_1)$	5.907 .203	4.956 .217
	$+E^*x\beta$	5.103 .157	3.603 (3.452) .166 (0.162)

and Fisher data for background speakers and SRE-04 for estimating WSCM. However, the trend is different on SRE-06 data. The configuration for the best performance on SRE-05 actually gives the worst performance on SRE-06. In addition, the results show a strong dependence on the choice of background data, such that significant improvements are obtained with SRE-04 data over Switchboard+Fisher data. The best performance on SRE-06 is obtained using SRE-04 data for background speakers and for the WSCM. The free parameters as optimized for SRE-05 test data and SRE-04 background and ISV data were $\alpha = 0.275$ and $\beta = 0.7$. Values optimized for other background and ISV data sets were within 20% of these settings.

We analyze the results further by dividing the performance of WCCN features into those coming from normalized projections on E (referred to as V_1) and then appending these to the weighted reconstructed vector obtained from E^* (referred to as V_2). Table VI shows the results corresponding to the results shown in the third and sixth rows of Table V (best performance on SRE-05 and SRE-06). It also shows the best possible cheating performance obtained with parameters chosen on SRE-06 data (last column, in parentheses). The results show that V_1 performs better than the baseline on SRE-05 and the performance is further improved by adding V_2 . However, the performance of V_1 does not generalize to SRE-06, especially when Switchboard and Fisher data is used to model background speakers. We hypothesize that the lack of generalization is due to differences in the data collections for SRE-04 and SRE-06, e.g., the fact that SRE-04 consists mostly of native speakers, whereas SRE-06 has a significant proportion of nonnative speakers. Further experimentation is needed to test and refine this hypothesis.

Table VII shows the results using NAP on different data sets for background speakers and for estimating WSCM. Note that the results for the N/A row are slightly different from Table V

TABLE VII

NAP RESULTS (N/A INDICATES BASELINE RESULTS WITHOUT NAP). NUMBERS IN BOLD SHOW THE BEST PERFORMANCE ON EACH TEST SET. NUMBERS IN PARENTHESES IN THE LAST ROW SHOW RESULTS IF PARAMETERS ARE TUNED ON SRE-06 INSTEAD OF ON SRE-05.

Background data	ISV data	SRE-05	SRE-06
		1-side	1-side
SWB+Fisher	N/A	5.866 .190	4.641 .225
	SRE-03	5.909 .169	4.423 .206
	SRE-04	5.470 .158	3.999 .197
SRE-04	N/A	6.189 .199	4.312 .197
	SRE-03	5.744 .171	3.831 .179
	SRE-04	5.664 .163	3.614 (3.567) .170 (0.167)

because of implementation differences.⁵

The numbers in parentheses in the last rows of Table V and Table VII show the cheating performance on SRE-06, using optimal parameters for that data set. These results show a different trend. NAP gives better improvement on SRE-06 than on SRE-05. In addition, NAP seems to be more sensitive to the choice of data for WSCM than to the choice of data for background speakers. The best performance for NAP on SRE-05 is obtained with the SWB+Fisher/SRE-04 configuration for background and ISV estimation, respectively, but the best performance on SRE-06 is obtained with SRE-04/SRE-04.

Comparison of WCCN and NAP results shows a difference in the best configurations for SRE-05 and SRE-06. It shows the importance of matched setups and that the worst-case mismatch in the configuration gives only a small improvement in performance. The comparison also shows the dependence of these techniques on choice of background corpus and data used for WSCM. Comparison of the cheating performance shows that WCCN suffers more from over-training than NAP. This is not surprising as the former uses more parameters. However, the best performance for both methods is obtained with the SRE-04/SRE-04 configuration, where both methods give comparable results.

G. Combining NAP with WCCN

We can explore simple combinations of NAP and WCCN. The idiosyncrasies of these approaches are as follows: NAP uses a very simple, binary weighting for the eigenvectors. WCCN models the subspace spanned by the eigenvectors corresponding to nonzero eigenvalues separately, and uses a complex weighting for the eigenvectors. Separating the subspaces is based on previous work [31], where it was shown

⁵In Table V, the results are obtained by running the baseline system without WCCN. In Table VII, the results are obtained by removing zero eigenvectors (default case for NAP). As a part of the NAP procedure, the global mean is subtracted from the original features for this default case, which leads to small numerical differences in features and scores. However, the difference between the two N/A results is not significant at the 95% confidence level. The purpose of showing a different result for NAP is to validate the experimental procedure by verifying that the default case gives essentially the same result as the baseline.

that, for cepstral features, it is advantageous to model these subspaces separately. However, the weighting proposed in [31] was simpler than the one used in WCCN (cf. the discussion in Section III-D).

Two combinations of NAP and WCCN were devised, as follows:

- 1) **NAP→WCCN**: Obtain the best NAP result, separate the subspaces as WCCN (ignore leading N eigenvectors) and apply simple weights as suggested in [31].
- 2) **WCCN→NAP**: Obtain the best WCCN result and modify the weighting so the first few eigenvectors are set to zero.

Preliminary experiments with these combinations do not show a significant improvement over the best NAP and WCCN results, but they do show interesting trends. In the NAP→WCCN combination, the results show that separating the spaces does not give any advantage over combining them. It also shows the same trend as WCCN results whereby the performance of the features that are projections onto the eigenvectors (V_1) does not generalize from SRE-05 to SRE-06. In the WCCN→NAP combination, the results do not change significantly if the weights of the leading eigenvectors are set to zero. This shows that the proposed weighting scheme is optimal in the given setup. However, there is a potential for pursuing different functional forms (e.g., sigmoid) for more compact and generic weightings.

H. Comparison with other cepstral systems

It is informative to compare the effectiveness of intersession variability compensation on the MLLR-SVM system with that on comparable cepstral speaker verification approaches. For this purpose we assembled two baseline systems. The first is a cepstral GMM-UBM as described in Section II-G, but without TNORM. Session variability compensation on this system was performed using factor analysis [9]. The second baseline was a Gaussian supervector system based on the cepstral GMM, following the approach of [13]. For this system, as for the MLLR-SVM, NAP was used for intersession variability compensation. The number of nuisance dimensions was optimized on SRE-05 data, and was set at 128 for the cepstral GMM and supervector systems, 15 for the MLLR-SVM (1-side) system, and 16 for the MLLR-SVM (8-side) system. Background and ISV statistics were derived from SRE-04 data. Results on SRE-06 data are summarized in Table VIII.

The results show that all three systems benefit significantly from intersession variability modeling, and that the improvement is inversely related to the performance before compensation (poorer systems improve more). The fact that the cepstral baseline systems improve more by ISV compensation correlates with their larger optimal number of nuisance dimensions, which in turn reflects the fact that the MLLR-SVM, by its design, is less affected by some of the factors contributing to ISV, such as phonetic content. However, even after ISV compensation, the MLLR-SVM system has an advantage over the other two cepstral systems. For 1-side training, the supervector system has second-best performance, whereas for

TABLE VIII

SPEAKER VERIFICATION PERFORMANCE OF THREE SYSTEMS WITH AND WITHOUT INTERSESSION VARIABILITY COMPENSATION. THE MLLR-SVM (1-SIDE) RESULTS CORRESPOND TO LAST COLUMN OF THE FORTH AND SIXTH ROWS OF TABLE VII.

	SRE-06 1-side		SRE-06 8-side	
	W/o ISV	With ISV	W/o ISV	With ISV
Cepstral GMM	6.150 .276	4.747 .216	4.576 .183	2.793 .107
Supervector SVM	5.556 .255	4.207 .200	4.776 .175	3.333 .172
MLLR-SVM	4.312 .197	3.614 .170	2.837 .109	2.638 .096

8-side training the cepstral GMM is a close second to the MLLR-SVM.

IV. CONCLUSION

We have proposed a speaker recognition approach based on SVM modeling of the speaker adaptation transforms found in modern speech recognition systems. By combining MLLR transforms for multiple recognition models and phone classes we obtain a system that rivals or exceeds the accuracy of state-of-art speaker verification with frame-cepstrum-based features and GMM or SVM modeling. Furthermore, the MLLR system gives additional gains in combination with cepstral systems. A variant of this approach can be used with a phone-loop recognition model, without word recognition, yielding a method that has low computational overhead, is easily ported to many languages, and still yields high-accuracy speaker recognition, as we have shown on both English and Arabic speaker verification tasks.

Further, we compared two techniques for compensating for intersession variability (WCCN and NAP) as applied to our MLLR-SVM system. Both techniques model intersession variability as a within-speaker covariance matrix and weight the resulting eigenvectors to minimize the variability. The results show that NAP is more sensitive to the choice of data for obtaining covariance statistics, and WCCN is more sensitive to the choice of the background set. WCCN gives the best performance on SRE-05 (the tuning set) but does not generalize as well to SRE-06 as NAP. Although WCCN requires more parameters and has generalization issues, it still performs comparably to NAP under the best configuration. Furthermore, we have explored combinations of these two techniques. Our preliminary results show limited gains, but there is a potential for using a functional form for the weightings of eigenvectors that will be more compact and more general.

In summary, the combination of MLLR-SVM modeling and session variability normalization yields speaker verification performance that is comparable to the best reported results on NIST evaluation data.

ACKNOWLEDGMENTS

The authors thank Andrew Hatch from ICSI for fruitful discussions and assistance with WCCN experiments, Pavel Matejka from Brno University for assistance with NAP experiments, Ramana Rao Gadde for clarifications concerning

SRI's implementation of MLLR adaptation, and Robbie Vogt from Queensland University of Technology for help with implementing NAP and the supervector system. We also thank the three anonymous reviewers whose suggestions greatly improved the paper.

REFERENCES

- [1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models", *Digital Signal Processing*, vol. 10, pp. 181–202, 2000.
- [2] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 161–164, Orlando, FL, May 2002.
- [3] D. Reynolds, "Channel robust speaker verification via feature mapping", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 53–56, Hong Kong, Apr. 2003.
- [4] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems", *Digital Signal Processing*, vol. 10, pp. 42–54, Jan. 2000.
- [5] A. Park and T. J. Hazen, "ASR dependent techniques for speaker identification", in J. H. L. Hansen and B. Pellom, editors, *Proceedings of the International Conference on Spoken Language Processing*, pp. 1337–1340, Denver, Sep. 2002.
- [6] D. E. Sturim, D. A. Reynolds, R. B. Dunn, and T. F. Quatieri, "Speaker verification using text-constrained Gaussian mixture models", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 677–680, Orlando, FL, May 2002.
- [7] K. Boakye and B. Peskin, "Text-constrained speaker recognition on a text-independent task", in *Proceedings Odyssey-04 Speaker and Language Recognition Workshop*, Toledo, Spain, May 2004.
- [8] S. Kajarekar, N. Malayath, and H. Hermansky, "Analysis of speaker and channel variability in speech", in *Proceedings IEEE Workshop on Speech Recognition and Understanding*, Keystone, Colo., Dec. 1999.
- [9] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Factor analysis simplified", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 637–640, Philadelphia, Mar. 2005.
- [10] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 629–632, Philadelphia, Mar. 2005.
- [11] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition", in *Proceedings of the International Conference on Spoken Language Processing*, pp. 1471–1474, Pittsburgh, PA, Sep. 2006.
- [12] L. Gillick, J. Baker, J. Bridle, M. Hunt, Y. Ito, S. Lowe, J. Orloff, B. Peskin, R. Roth, and F. Scattone, "Application of large vocabulary continuous speech recognition to topic and speaker identification using telephone speech", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 471–474, Minneapolis, Apr. 1993.
- [13] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification", *IEEE Signal Processing Letters*, vol. 13, pp. 308–311, May 2006.
- [14] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers", in *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pp. 487–493, 1999.
- [15] P. Moreno and P. Ho, "A new SVM approach to speaker identification and verification using probabilistic distance kernels", in *Proceedings of the 8th European Conference on Speech Communication and Technology*, pp. 2965–2968, Geneva, Sep. 2003.
- [16] A. Martin, D. Miller, M. Przyboccki, J. Campbell, and H. Nakasone, "Conversational telephone speech corpus collection for the NIST speaker recognition evaluation 2004", in *Proceedings 4th International Conference on Language Resources and Evaluation*, pp. 587–590, Lisbon, May 2004.
- [17] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines", in *Advances in Neural Information Processing Systems 16*, 2004.

- [18] S. S. Kajarekar, L. Ferrer, E. Shriberg, K. Sonmez, A. Stolcke, A. Venkataraman, and J. Zheng, "SRI's 2004 NIST speaker recognition evaluation system", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 173–176, Philadelphia, Mar. 2005.
- [19] A. Stolcke, H. Franco, R. Gadde, M. Graciarena, K. Precoda, A. Venkataraman, D. Vergyri, W. Wang, J. Zheng, Y. Huang, B. Peskin, I. Bulyko, M. Ostendorf, and K. Kirchhoff, "Speech-to-text research at SRI-ICSI-UW", in *DARPA RT-03 Workshop*, Boston, May 2003, <http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/sri+-rt03-stt.pdf>.
- [20] N. Kumar, *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*, PhD thesis, Johns Hopkins University, Baltimore, 1997.
- [21] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, pp. 1747–1750, Istanbul, June 2000.
- [22] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 339–341, Atlanta, May 1996.
- [23] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition", *Computer Speech and Language*, vol. 12, pp. 75–98, Apr. 1998.
- [24] H. Jin, S. Matsoukas, R. Schwartz, and F. Kubala, "Fast robust inverse transform SAT and multi-stage adaptation", in *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, pp. 105–109, Lansdowne, VA, Feb. 1998, Morgan Kaufmann.
- [25] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of HMMs", *Computer Speech and Language*, vol. 9, pp. 171–186, 1995.
- [26] V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker adaptation using constrained reestimation of gaussian mixtures", *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 357–366, 1995.
- [27] L. Ferrer, E. Shriberg, S. S. Kajarekar, A. Stolcke, K. Sönmez, A. Venkataraman, and H. Bratt, "The contribution of cepstral and stylistic features to SRI's 2005 NIST speaker recognition evaluation system", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 101–104, Toulouse, May 2006.
- [28] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman, "MLLR transforms as features in speaker recognition", in *Proceedings of the 9th European Conference on Speech Communication and Technology*, pp. 2425–2428, Lisbon, Sep. 2005.
- [29] T. Joachims, "Making large-scale SVM learning practical", in B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods—Support Vector Learning*, pp. 41–56. MIT-Press, 1999.
- [30] A. O. Hatch, A. Stolcke, and B. Peskin, "Combining feature sets with support vector machines: Application to speaker recognition", in *Proceedings IEEE Workshop on Speech Recognition and Understanding*, pp. 75–79, San Juan, Puerto Rico, Nov. 2005.
- [31] S. S. Kajarekar, "Four weightings and a fusion: A cepstral-SVM system for speaker recognition", in *Proceedings IEEE Workshop on Speech Recognition and Understanding*, pp. 17–22, San Juan, Puerto Rico, Nov. 2005.
- [32] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Improvements in factor analysis based speaker verification", in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 113–116, Toulouse, May 2006.
- [33] R. Vogt, B. Baker, and S. Sridharan, "Modelling session variability in text-independent speaker verification", in *Proceedings of the 9th European Conference on Speech Communication and Technology*, pp. 3117–3120, Lisbon, Sep. 2005.
- [34] A. Solomonoff, C. Quillen, and I. Boardman, "Channel compensation for SVM speaker recognition", in *Proceedings Odyssey-04 Speaker and Language Recognition Workshop*, Toledo, Spain, May 2004.
- [35] B. Schoelkopf, A. J. Smola, and K.-R. Mueller, "Kernel principal component analysis", in B. Schoelkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pp. 327–352. MIT Press, 1998.
- [36] L. Burget, P. Matejka, P. Shwarz, O. Glembek, M. Karafiat, J. Cernocky, and F. Grezl, "NIST speaker recognition evaluation 2006", NIST 2006 Speaker Recognition Workshop, July 2006.



Andreas Stolcke (M'95, SM'05) received a Ph.D. in Computer Science from the University of California at Berkeley in 1994. He is a Senior Research Engineer at SRI International, Menlo Park, CA, and at the International Computer Science Institute, Berkeley, CA. His research interests are in applying novel modeling and learning techniques to speech recognition, speaker identification, and natural language processing. He is also the author of a widely used open-source toolkit for statistical language modeling.



Sachin S. Kajarekar received a Ph.D. degree from OGI School of Science and Engineering, Hillsboro, Oregon, in 2002. He is a Research Engineer at SRI International, Menlo Park, CA. His research interests are in feature extraction for speaker recognition for speech and speaker recognition.



Luciana Ferrer received a B.S. degree from the University of Buenos Aires, Buenos Aires, Argentina, in 2001. She is currently pursuing a Ph.D. degree at Stanford University, Stanford, CA. Before starting her studies at Stanford University, she worked for three years at SRI International as a research scholar. Her research interests include machine learning and information theory as applied to speech processing problems.



Elizabeth Shriberg received the Ph.D. degree from the University of California, Berkeley, in 1994. She is a Senior Researcher at both SRI International, Menlo Park, CA, and the International Computer Science Institute, Berkeley, CA. Her interest is in how people really talk. Her work aims to combine linguistic knowledge with corpora and techniques from speech and speaker recognition, to advance both scientific understanding and recognition technology.