Lattice-based MLLR for Speaker Recognition

Marc Ferràs[1], Claude Barras[1,2] and Jean-Luc Gauvain[1*]
[1]LIMSI-CNRS, BP 133, 91403, Orsay, France
[2]Univ Paris-Sud, F-91405, Orsay, France
{ferras,barras,gauvain}@limsi.fr

ABSTRACT

Maximum-Likelihod Linear Regression (MLLR) transform coefficients have shown to be useful features for text-independent speaker recognition systems. These use MLLR coefficients computed on a Large Vocabulary Continuous Speech Recognition System (LVCSR) as features and Support Vector machines(SVM) classification. However, performance is limited by transcripts, which are often erroneous with high word error rates (WER) for spontaneous telephone speech applications. In this paper, we propose using lattice-based MLLR to overcome this issue. Using word-lattices instead of 1-best hypotheses, more hypotheses can be considered for MLLR estimation and, thus, better models are more likely to be used. As opposed to standard MLLR, language model probabilities are taken into account as well. We show how systems using lattice MLLR outperform standard MLLR systems in the Speaker Recognition Evaluation (SRE) 2006. Comparison to other standard acoustic systems is provided as well.

Index Terms— Speaker recognition, MLLR, lattice

1. INTRODUCTION

Recent approaches to text-independent speaker recognition focus on finding alternative features to cepstral coefficients. These coefficients are typically sensitive to many factors of acoustic and linguistic variability that can mask speaker information, namely message and channel variability[1]. Modeling these undesired variabilities for its compensation is a common way to reduce their intra-speaker variance, thus improving discrimination among speakers. Conversely, increasing inter-speaker variance directly, i.e. finding more speaker-relevant and robust features, can be seen as leading to the same goal.

The time-dependent nature of cepstral coefficients is a drawback for certain modeling paradigms such as Support Vector Machines (SVM). A common approach to overcome this issue is mapping the variable-length sequence into a fixed-length vector, where each of its coefficients operates over the whole sequence. These fixed-length feature vectors, which are often high-dimensional, are aimed at representing speaker-related characteristics in a more compact way. Polynomials and radial basis functions have been used as mappings in [2]. However, much more complex mappings can result in interesting fixed-length feature vectors. In this sense, Gaussian Mixture Model (GMM) [3] Supervectors and Maximum-Likelihood Linear Regression (MLLR) transforms as features [4, 5, 6, 7] have been proposed in the last years. These high-dimensional feature vectors lie in very sparse vector spaces. Support Vector Machines (SVM) have shown significant robustness under these assumptions [8], becoming a predominant modeling framework in speaker recognition.

MLLR transforms as features were first used in speaker recognition in [4]. Systems estimate one or several MLLR transforms using speaker-independent Gaussian models, either GMMs or Hidden Markov Models (HMM), and the adaptation data for a speaker. These affine transforms capture the difference between the speaker-independent model and the speaker-adapted model. The resulting matrix coefficients are used as features. Systems using GMM for MLLR computation do not require transcripts and are language-independent whereas systems using phonemic HMMs from Large Vocabulary Continuous Speech Recognition (LVCSR) systems do. However, performance of the latter is considerably higher. Other similar approaches using constrained MLLR (CMLLR) [9] have also been proposed in the literature [6, 7].

When used in a LVCSR system, MLLR adaptation requires transcripts as well as a pronounciation lexicon. Speech data are aligned against the acoustic models corresponding to the given transcript. However, automatic transcripts on telephone speech tend to be errorful with typical error rates higher than 20%. In such situation we take the risk of not using the correct acoustic models by using the 1-best hypothesis only. In this paper, we use the word lattices produced on a first-pass decoding of a LVCSR system as reference to compute the MLLR transforms, thus accounting for the erroneous transcripts. The resulting MLLR transforms are then used as features by rearranging their coefficients into vector form.

This paper is organized as follows: Section 2 introduces MLLR and Section 3, lattice MLLR. Section 4 describes feature extraction. The experimental set-up as well as the evaluation task are detailed in Section 5 and system description in Section 6. Section 7 presents and discusses NIST Speaker Recognition Evaluation (SRE) 2006 results of MLLR-based and other acoustic-level systems. Conclusions are given in Section 8.

2. MAXIMUM LIKELIHOOD LINEAR REGRESSION (MLLR)

Maximum-Likelihood Linear Regression (MLLR) [10] is an acoustic adaptation technique typically used for speaker adaptation purposes in HMM-based speech recognition systems. The idea behind is to adapt the parameters of an HMM using an affine transform given T speaker-dependent observation vectors, $\mathbf{o}_T$, so as to maximize its log-likelihood with respect to the adapted model as

$$\hat{\theta}^* = \arg\max_{\hat{\theta}} \log p(\mathbf{o}_T|\hat{\theta}) \quad (1)$$

*This work has been partially financed by OSEO under the Quaero program.

where $\hat{\theta}^*$ are the parameters of the adapted model, $\hat{\boldsymbol{\mu}}_s = \mathbf{A}\boldsymbol{\mu}_s + \mathbf{b}_s$ and $\hat{\boldsymbol{\Sigma}}_s = \boldsymbol{\Sigma}_s$, i.e. mean vector and covariance matrix, respectively, for a given state or tied-state s. In this framework, mean adaptation is performed by means of an affine transform while covariance matrices are not adapted.

A preliminary step to solve (1) is to align the observed feature vectors against the model states. This is performed by force-aligning the data against the transcripts, the words of which can be decomposed into model-level states by means of the pronounciation lexicon. The optimization process is typically performed using the Expectation-Maximization algorithm [10], although other approaches have also been proposed [11].

3. LATTICE MLLR

The MLLR approach, as described in Section 2, relies on the transcripts for state alignment, which plays an important role in chosing the models used for MLLR adaptation. However, transcripts are subject to errors, specially in those tasks involving spontaneous and low-quality speech. Word Error Rates (WER) for these applications tend to be high, typically over 20%.

Errors in the transcripts can be accounted for by modeling the uncertainty in the state alignment. In this sense, (1) can be restated as the maximization of the expected conditional log-likelihood given all possible alignments $\mathbf{S}$, from observation sequence $\mathbf{o}_T$ to state sequence $\mathbf{s_T}$ as

$$\hat{\theta}^* = \arg\max_{\hat{\theta}} \sum_{\mathbf{s}_T \in \mathbf{S}} p(\mathbf{s_T}|\mathbf{o}_T, \theta) \log p(\mathbf{o}_T, \mathbf{s_T}|\hat{\theta}) \qquad (2)$$

where $p(\mathbf{s_T}|\mathbf{o}_T, \theta)$ is the probability of aligning the observation sequence $\mathbf{o}_T$ into state sequence $\mathbf{s}_T$ using the non-adapted model, θ. In standard MLLR, these probabilities are set to 1 for the chosen alignment and 0 for all other paths, ignoring all paths except that involving the 1-best hypothesis, which reduces (2) to (1). The formulation in (2) also allows to include cross-word transition probabilites in the MLLR estimation, thus using information provided by the language model as well. Furthermore, since one observation vector can be mapped to several states, more speech data is used overall for estimating each transforms. Conversely, more transforms can be used with the same amount of data per transform.

Based on this framework, lattice-based MLLR [12, 13] uses the word-lattice output of an ASR system obtained in a first-pass decoding to estimate MLLR transforms. The word-level graph is collapsed down to a model-level graph using the pronounciation variants in the lexicon to eventually find all possible alignment probabilities, $p(\mathbf{s_T}|\mathbf{o}_T, \theta)$. Thus, the transition probabilities are specified by both left-to-right within-word models and the arcs of the word lattice. The most likely alignments have a strong effect on the likelihood function whereas the least likely alignments are given a smaller weight and can be eventually pruned to reduce computation load. A threshold can also be set at the Gaussian posterior probability level to filter out unlikely Gaussians.

4. MLLR FEATURE EXTRACTION

Following the approach presented in [4, 5], MLLR transforms are estimated for each speaker of interest using an HMM-based LVCSR system. Depending on the amount of speech data available for adaptation, the acoustic space can be split into several acoustic classes, static or dynamically derived, which result in several MLLR transforms. Using many classes results in a finely represented phonetic space but less speech data is available for each class-dependent transform. All MLLR matrix coefficients are eventually re-arranged into vector form to make up a single high-dimensional feature vector that characterizes the speaker we have adapted for.

5. EXPERIMENTAL SETUP

We evaluated both MLLR and lattice MLLR techniques in the NIST Speaker Recognition Evaluation 2006[1]. SRE'06 consists of conversational telephone speech data involving multiple languages, dialects as well as multiple acoustic conditions. However, we targeted the common condition, which involves English language trials only, the main reason behind being that our LVCSR system is developed for English. Data consists of 5-minute-long segments with about two minutes of average effective speech per conversation side[2]. 816 (354 male / 462 female) speakers are available for model training and 3735 (1606 male / 2129 female) are for test, resulting in over 20000 cross-channel trials, with a proportion of 60 impostor speakers per true speaker. Mismatch in the acoustic channel as well as in the dialect is allowed.

6. SYSTEM DESCRIPTION

The LVCSR system used for computation of MLLR transforms is based on the LIMSI RT'04 LVCSR system [14]. This system was trained using Speaker Adaptive Training (SAT) on about 650 hours of speech data, including Switchboard I (4862 conversation sides), Switchboard II (2348 sides), Callhome (240 sides) and Fisher (6127 sides) corpora. The front-end was optimized for speaker recognition and it uses 15 PLP coefficients along with its Δ, $\Delta\Delta$ coefficients, Δ and $\Delta\Delta$ energies, feature mapping for channel compensation and feature warping. Acoustic models are gender-independent tied-state context-dependent triphones. Tied-states were found by means of a decision tree, resulting in 6100 tied-states with 32 Gaussians per state.

In the following, we present the rest of speaker recognition systems involved in our experiments. All of these use the same front-end used in the LVCSR system, except for segmentation. MLLR-based systems used forced-alignment for segmentation while other acoustic-based systems used a Speech Activity Detector (SAD) that considered only voiced regions in the speech signal.

MFCC-GMM system

The MFCC-GMM system is based on the GMM-UBM paradigm with diagonal covariance matrices trained using 3 iterations of MAP adaptation. The GMM is gender-dependent with 1536 Gaussians (512x3). For compensating inter-session variability of GMMs supervectors, we use hybrid Factor Analysis (FA) [15] where GMM are compensated at the model level and test segments are compensated in the feature domain. The channel-loading matrix was trained on the SRE'04 data and we used 40 dimensions for the channel subspace. Speaker GMM are compensated at the model level and test segments are compensated in the feature domain. Scores are T-norm normalized using 500 speech segments (250 males and 250 females) taken from SRE'04 data as well. We use a forward-backward scoring approach, where test speech is scored against the target model and target

[1] The NIST year 2006 speaker recognition evaluation plan, `http://www.nist.gov/speech/tests/spk/2006/`

[2] The common condition involves two sides, or alternatively, four wires.

speech is scored against the test model. Therefore, we obtain 2 scores per trial which are averaged to give one final score per trial.

SVM-based systems

All SVM-based systems use the same SVM set-up and pre-processing, differring in the features used. For pre-processing, we apply Nuisance Attribute Projection (NAP) to project out the subspace of maximum intra-speaker variability, thus compensating inter-session variability. We use the NIST SRE'04 dataset as NAP training data. An affine transform maps each feature component into the range [-1/$\sqrt{D}$,1/$\sqrt{D}$], D being the dimension of the feature vector, so that only normalized dot products are processed by the SVM.

We use SVMTorch[3] with a linear kernel set-up for SVM classification. The impostor speaker set consists of 2243 speech segments from the NIST SRE04 training data plus 4854 speech segments from the Switchboard I corpus. All of them are in English language and have a minimum duration of 10 seconds of speech (after forced-alignment). This configuration allows all SVM-based systems to share the same impostor data, as transcripts[4] are available for all of the 7097 segments. We use the forward-backward scoring approach as in the MFCC-GMM system.

MFCC-SVM system

The MFCC-SVM system is based on the GLDS kernel[2], using a third-order polynomial feature extraction scheme. The resulting features were variance normalized and averaged over the whole segment to obtain a single 20824-dimensional vector.

GSV-SVM system

The GSV-SVM system uses Gaussian mean supervectors of a GMM as features. GMMs are adapted using MAP adaptation from a gender-dependent UBM. We use 512 Gaussians and variance-normalization.

MLLR-SVM systems

MLLR-SVM systems use either standard MLLR or lattice MLLR transforms as features as described in Sections 2 and 3, respectively. We experimented with two to four MLLR classes (non-speech/speech, non-speech/consonants/vowels and non-speech/consonants/vowels1/vowels2). These classes were manually derived based on phonological cues. MLLR tranforms for the non-speech class were not used as they were assumed not to carry any relevant speaker information. Thus, we obtained a maximum of three transforms using four acoustic classes. NAP set-up, feature normalization and modeling were kept the same as in MFCC-SVM systems.

7. RESULTS

Our experiments were set to compare systems using different MLLR approaches, i.e. lattice MLLR vs. standard MLLR. We generated 1-best hypotheses and lattices using our LVCSR system. Table 1 shows Detection Cost Function[5] (DCF) for the a posteriori optimal threshold, i.e. Minimum Detection Cost (MDC), and Equal Error Rate (EER) results obtained for the common condition in the SRE'06 evaluation for all systems. Matched-pairs statistical significance tests[6] were also performed for some pairs of systems to check reliability of shown improvements.

Regarding MLLR-SVM systems (second block in Table 1), we observe a trend to lower error rates as we use more transforms. However, error rates increase again when using four acoustic classes (MLLR-SVM 3t), possibly due to either the lack of data for reliably estimating transforms or the choice of the static class definitions (two subsets of vowels instead of only one). This trend is also observed for systems using word-lattices (third block in Table 1), but the performance loss using three transforms rather exhibits a saturation effect. We believe such effect is due to observation data sharing among several Gaussians when using word-lattices. Improvement of Lat. MLLR-SVM 3t vs. Lat. MLLR-SVM 2t fails to be significant, as shown in the last row of Table 2. Systems using Lattice MLLR outperform standard MLLR regardless of the number of transforms, although not significantly when using three transforms (see Table 2. For systems using two transforms, the performance increase is consistent for most of the operating points and around the MDC region especially, as shown in Fig. 1(a). Relative gains in MDC of 8% are found for MLLR-SVM 1t vs. MLLR-SVM 2t, around 5% for MLLR-SVM 3t.

Overall, MLLR-SVM systems outperform all other acoustic-level systems both for MDC and EER operating points, as illustrated in Fig. 1(b). Standard MLLR and lattice MLLR using two transforms obtain relative gains of 5% and 13% MDC respectively when compared to MFCC-GMM, which is the best performing amongst non-MLLR systems. In terms of EER, gains of 3.5% to 4.5% in absolute terms are observed.

System	MDC	EER (%)
MFCC-GMM	0.0176	3.72
MFCC-SVM	0.0188	4.09
GSV-SVM	0.0183	3.67
MLLR-SVM 1t	0.0189	4.45
MLLR-SVM 2t	0.0166	**3.63**
MLLR-SVM 3t	**0.0162**	4.18
Lat. MLLR-SVM 1t	0.0174	4.23
Lat. MLLR-SVM 2t	**0.0152**	**3.63**
Lat. MLLR-SVM 3t	0.0153	3.68

Table 1: MDC and EER for the MLLR-SVM and other acoustic-level systems on the SRE'06 evaluation data (key v11). Lowest DCF and EER in a series of results are shown in boldface.

8. CONCLUSIONS

Lattice MLLR was proposed as a means to deal with erroneous transcripts in text-independent speaker recognition systems. Systems based on this approach exhibited consistent gains over systems based on standard MLLR in the NIST SRE06 evaluation. Using 4 acoustic classes, lattice MLLR helped mitigate performance

[3]SVMTorch: a Support Vector Machine for Large-Scale Regression and Classification Problems - `http://www.idiap.ch/learning/SVMTorch.html`

[4]Manual transcripts for Switchboard I, ASR transcripts for SRE'04.

[5]The NIST SRE 2006 campaign defines DCF as a weighted sum of false alarm and miss errors which becomes $P_{Miss} + 9.9 \times P_{FalseAlarm}$ after normalization.

[6]Assuming systems A and B made M different binary decisions of which N are correct for B, the probability that one system improved performance with respect to the other by chance, i.e. at least N correct decisions be randomly made, can be modeled by the cumulative Binomial distribution $2P(k \geq N|M) = 2\sum_{k=N}^{M} \binom{M}{k} p^k (1-p)^{M-k}$, where correct decisions have a probability of $p = 0.5$.

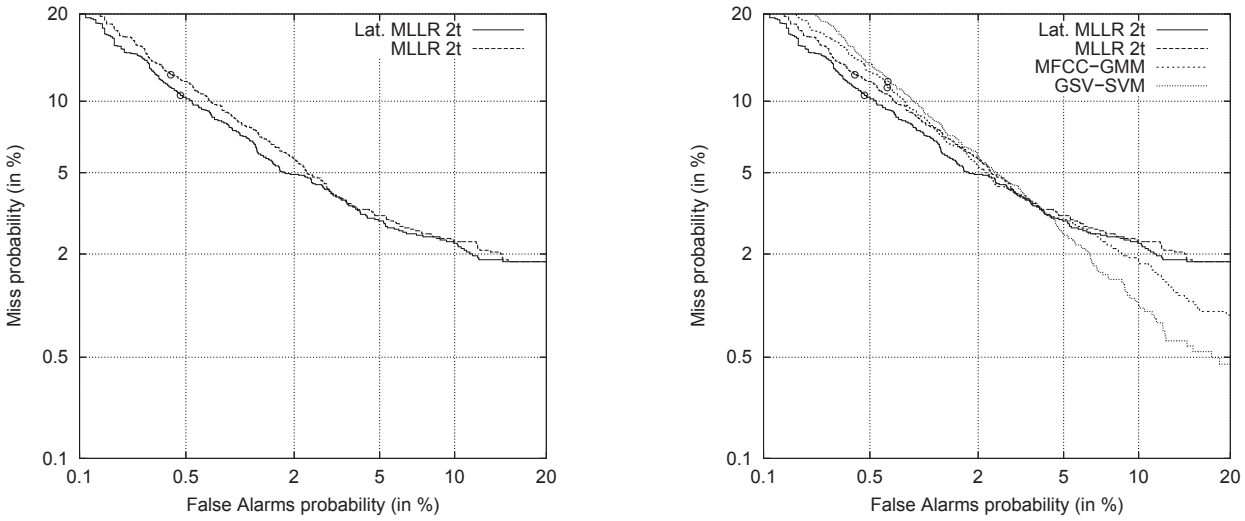

Figure 1: DET curves for MLLR individual systems (left, a): Lattice MLLR-SVM 2t vs. MLLR-SVM 2t. DET curves for other individual systems (right, b): Lattice MLLR-SVM 2t, MLLR-SVM 2t, MFCC-GMM and GSV-SVM. MDC operating points are shown as dots.

From System	To System	P-Value
MLLR-SVM 1t	Lat. MLLR-SVM 1t	0.002
MLLR-SVM 2t	Lat. MLLR-SVM 2t	0.004
MLLR-SVM 3t	Lat. MLLR-SVM 3t	**0.077**
Lat. MLLR-SVM 1t	Lat. MLLR-SVM 2t	0.002
Lat. MLLR-SVM 2t	Lat. MLLR-SVM 3t	**0.87**

Table 2: Significance levels for several system comparisons. Weak significance is reached for p-values between 0.05 and 0.01 and no significance for p-values above 0.05, both shown in boldface.

degradation due to the lack of data assigned to each transform. Both MLLR-based approches outperformed all other acoustic-level systems, including MFCC-GMM and GSV-SVM, currently considered as state-of-the-art.

REFERENCES

[1] S. Kajarekar, N. Malayath, and H. Hermansky, "Analysis of Speaker and Channel Variability in Speech," *Proceedings of the IEEE Workshop on Speech Recognition and Understanding*, December 1999.

[2] W. M. Campbell, "Generalized Linear Discriminant Sequence Kernels for Speaker Recognition," *Proceedings of IEEE Conference on Audio Speech and Signal Processing*, 2002.

[3] 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, no. 5, pp. 308–311, 2006.

[4] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman, "MLLR Transforms as Features in Speaker Recognition," *Proceedings of Eurospeech*, pp. 2425–2428, September 2005.

[5] A. Stolcke, L. Ferrer, and S. Kajarekar, "Improvements in MLLR-Transform-Based Speaker Recognition," *Proceedings of the IEEE Speaker Odyssey*, June 2006.

[6] M. Ferràs, C. C. Leung, C. Barras, and J-L Gauvain, "Constrained MLLR for Speaker Recognition," *Proceedings of IEEE Conference on Audio Speech and Signal Processing*, April 2007.

[7] M. Ferràs, C.C. Leung, C. Barras, and J.L. Gauvain, "MLLR Techniques for Speaker Recognition," in *Proceedings of IEEE Speaker Odyssey*, January 2008.

[8] Y. Yang and X. Liu, "A re-examination of text categrorization methods," in *ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 42–49.

[9] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 357–366, September 1995.

[10] C. J. Leggetter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[11] S. V. Balakrishnan, "Fast incremental adaptation using maximum likelihood regression and stochastic gradient descent," in *Proceedings of Eurospeech*, February 2003.

[12] M. Padmanabhan, G. Saon, and G. Zweig, "Lattice-based unsupervised mllr for speaker adaptation," in *Proceedings of the ISCA ITRW ASR2000*, 2000, pp. 128–131.

[13] L.F. Uebel and P.C. Woodland, "Improvements in linear tranformation based speaker adaptation," in *Proceedings of IEEE Conference on Audio Speech and Signal Processing*, April 2001.

[14] R. Prasad, S. Matsoukas, C.-L. Kao, J. Ma, D.-X. Xu, T. Colthrust, O. Kimball, R. Schwartz, J.-L. Gauvain, L. Lamel, H. Schwenk, G. Adda, and F. Lefevre, "The 2004 BBN/LIMSI 20xRT English Conversational Telephone Speech Recognition System," *Proceedings of Interspeech*, 2005.

[15] D. Matrouf, N. Scheffer, B. Fauve, and J.F. Bonastre, "A Straightforward and Efficient Implementation of the Factor Analysis Model for Speaker Verification," in *Proceedings of INTERSPEECH*, August 2007.