interspeech 2008 incorporating SST 08

Parallelized Factor Analysis and Feature Normalization for Automatic Speaker Verification

Jun Luo, Cheung-Chi Leung, Marc Ferras and Claude Barras

Spoken Language Processing Group
LIMSI-CNRS, France
{luo,ccleung,ferras,barras}@limsi.fr

Abstract

Factor analysis (FA) is one of the key advances presented in recent speaker verification evaluations. This technique is able to successfully remove session variability effects and it is currently used in many state-of-the-art automatic speaker verification systems. This paper addresses several practical issues in using an FA model in order to speed up model training and to achieve good performance. A parallelized training algorithm as well as maximum-likelihood estimation are proposed for fast training. The front-end feature normalization techniques are also investigated in the context of FA model. We demonstrate that factor analysis is very robust, and can be successfully applied to various kinds of feature normalization. Moreover, the proposed parallelized MLE implementation speeds up the training procedure from several days to several hours without sacrificing the performance.

Index Terms: speaker verification, factor analysis, maximum-likelihood estimation, parallelization.

1. Introduction

One of the most important challenges in automatic speaker verification (ASV) is how to deal with the intersession variability. Under matched conditions, ASV systems usually perform well. This is not the case under mismatched conditions, where intersession variability compensation becomes a need. Factor analysis (FA) is one of the key innovations in recent NIST speaker verification evaluations being able to successfully deal with the intersession variability issue and is widely used in state-of-the-art ASV systems. The FA method models the intersession variability explicitly and the compensation can be performed either in feature domain or model domain, both are proved to be very effective [1].

There are also many compensation techniques in feature domain, such as feature cepstrum mean subtraction (CMS) [2], feature mapping [3] and feature warping [4], which address the variation issue and achieve good performance with a single technique. However, it is unclear if those techniques can still be useful with FA.

The contribution of this paper is to re-examine the feature normalization techniques in the context of FA modeling. CMS/feature mapping/feature warping are tested in combination with FA and compared to the feature where no normalizations at all is performed.

Another practical issue of using FA is that the training is usually very time-consuming, especially when more Gaussian mixtures and higher feature dimensions are used. To address this problem, we propose a parallelized implementation based on maximum-likelihood estimation (MLE) in this paper to support fast training. This estimation requires less storage of interim statistics without decreasing performance.

This paper is organized as follows: Section 2 introduces the factor analysis model and the session compensation approach. Section 3 presents the introduction of different feature normalization techniques. The parallelization implementation as well as Maximum Likelihood Estimation are introduced in section 4, the experimental results are given in section 5 and section 6 concludes the paper.

2. Session Variability Modeling and Hybrid Domain Session Compensation

GMM (Gaussian Mixture Models)-UBM (Universal Background Models) approach represents one of the most important widely used techniques in state-of-the-art speaker verification systems. The speaker models are estimated from a common GMM seed model (UBM). Usually this is done by MAP [5] adaptation only of the mean. For a set of speaker models only the mean vectors are different and the other parameters are shared with the UBM. In this case, each speaker is represented by a supervector constructed by concatenating all of the mean vectors.

The basic assumption in factor analysis model is that a speaker- and channel-dependent supervector can be decomposed into three different components: a speaker-session-independent component $\mathbf{m}$, a component which only depends on the speaker, and a component only depending on channel. The channel dependent component and speaker dependent component are assumed to be statistically independent and normally distributed.

A theoretical framework of factor analysis is proposed by Kenny in [6] and the reduced model, eigenchannel MAP is introduced in [7]. This model can be expressed as:

$$\mathbf{m}_{(h,s)} = \mathbf{m} + \mathbf{D}\mathbf{y}_s + \mathbf{U} \cdot \mathbf{x}_{(h,s)}, \quad (1)$$

where $\mathbf{m}_{(h,s)}$ is the mean of session-speaker dependent supervector (its dimension is $MD \times 1$, where M is the number of Gaussians, and D is the dimension of the feature) corresponding to session index h and speaker index s, $\mathbf{D}$ is a diagonal matrix, $\mathbf{y}_s$ is the speaker vector, $\mathbf{U}$ is the session variability matrix (a $MD \times R$ matrix) and $\mathbf{x}_{(h,s)}$ is the channel factor (a R dimensional vector). Each column in matrix $\mathbf{U}$ corresponds to one possible projection direction in channel-dependent space and all the R vectors account for most of the channel variability. Typically $R \ll MD$.

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

FA-based channel compensation can be implemented both in model domain and feature domain. The hybrid domain normalization strategy proposed in [8] is adopted in this paper. Given the speech data for target speaker, the session variability is subtracted from the target speaker model (and normalization models as well) to make the target model a "true speaker model" which is independent of session variability. The compensation in the testing data is performed at the frame level which can be regarded as a feature post-processing procedure. That is[1]:

$$\mathbf{m}_s = \mathbf{m} + \mathbf{D}\mathbf{y}_s; t' = t - \sum_{g=1}^{M} \gamma_g(t) \cdot \{\mathbf{U} \cdot \mathbf{x}_{h_{test}}\}_{[g]}, \quad (2)$$

where $\mathbf{m}_s$ is the target speaker model, t' is the compensated feature. The verification is based on the log-likelihood ration:

$$\log P(\mathcal{Y}|\mathbf{m}_{(h_{tar},s_{tar})}) - \log P(\mathcal{Y}|\mathbf{m}), \quad (3)$$

where $\mathbf{m}_{(h_{tar},s_{tar})}$ is the true speaker model and $\mathcal{Y}$ is the speech feature sequence after subtracting session effects.

3. Feature Normalization Techniques and Factor Analysis

Cepstrum Mean Subtraction (CMS) is an effective method for removing linear effects introduced by the communication channel from the speech signal, by subtracting the mean cepstrum from each feature in the duration of the utterance.

The objective of feature warping is to construct a more robust representation of the cepstrum feature distribution. It was found that cepstrum based feature vector warping using a Gaussian target distribution is an effective method of reducing the effects of mismatch.

Feature mapping uses the *a priori* information from a set of models trained in known conditions in order to map the feature vectors to a channel independent feature, and a data-driven technique has also been proposed to release the requirement of explicit identification and labeling of conditions.

All these techniques have been successfully used in ASV, however, if these techniques can well be applied with FA model, and which is the best is still not clear. The combination of different feature normalization techniques with FA as well as the best configuration are given in section 5.

4. Parallelized Factor Analysis Model Training

The training of session variability matrix $\mathbf{U}$ is quite time-consuming especially when many Gaussians are used with high dimensional vectors. Typically, the number of Gaussians is around several thousands. It takes long time to train a matrix $\mathbf{U}$ given thousands of sessions data without parallelization. However, this procedure could be parallelized in two stages:

- The estimation of each speaker and session dependent parameters;
- The estimation of transformation matrix $\mathbf{U}_{[g]}$ which corresponds to each Gaussian.

Moreover, if the maximum-likelihood estimation is performed, the storage of necessary statistics will be reduced and the training will be simplified.

[1]Notation: Let $\mathbf{A}$ be a $MD \times K$ matrix formed by concatenating vertically M matrices of dimesnions $D \times K$, we denote $\mathbf{A}_{[g]}$ the g^{th} matrix in $\mathbf{A}$.

4.1. Maximum-likelihood Estimation

We begin with the definition for general statistics. Let $\mathbf{N}_s$ and $\mathbf{N}_{(h,s)}$ be the vectors containing the zero order speaker-dependent and session-dependent statistics respectively, and $\mathbf{X}_s$ and $\mathbf{X}_{(h,s)}$, the first order statistics:

$$\mathbf{N}_{s[g]} = \sum_{t \in s} \gamma_g(t); \mathbf{N}_{(h,s)_{[g]}} = \sum_{t \in (h,s)} \gamma_g(t), \quad (4)$$

$$\mathbf{X}_{s[g]} = \sum_{t \in s} \gamma_g(t) \cdot t; \mathbf{X}_{(h,s)_{[g]}} = \sum_{t \in (h,s)} \gamma_g(t) \cdot t. \quad (5)$$

After that, the session effects and speaker effects are removed to give speaker dependent statistics $\overline{\mathbf{X}}_s$ and session dependent statistics $\overline{\mathbf{X}}_{(h,s)}$ respectively:

$$\begin{aligned} \overline{\mathbf{X}}_{s[g]} &= \mathbf{X}_{s[g]} - \sum_{h \in s} \mathbf{N}_{(h,s)_{[g]}} \cdot \{\mathbf{m} + \mathbf{U} \cdot \mathbf{X}_{(h,s)}\}_{[g]} \\ \overline{\mathbf{X}}_{(h,s)_{[g]}} &= \mathbf{X}_{(h,s)_{[g]}} - \{\mathbf{m} + \mathbf{D}\mathbf{y}_s\}_{[g]} \cdot \sum_{h \in s} \mathbf{N}_{(h,s)_{[g]}}, \end{aligned} \quad (6)$$

The statistics $\mathbf{L}_{(h,s)}$ and $\mathbf{B}_{(h,s)}$ are defined as follows:

$$\begin{aligned} \mathbf{L}_{(h,s)} &= \sum_{g \in \mathbf{UBM}} \mathbf{N}_{(h,s)_{[g]}} \cdot \{\mathbf{U}\}_{[g]}^t \cdot \mathbf{\Sigma}_{[g]}^{-1} \cdot \{\mathbf{U}\}_{[g]} \\ \mathbf{B}_{(h,s)} &= \sum_{g \in \mathbf{UBM}} \{\mathbf{U}\}_{[g]}^t \cdot \mathbf{\Sigma}_{[g]}^{-1} \cdot \overline{\mathbf{X}}_{(h,s)_{[g]}}. \end{aligned} \quad (7)$$

The speaker factor is estimated the same as MAP adaptation, but the channel factor follows the maximum likelihood eigen-decomposition (MLED) [9]. That is:

$$\begin{aligned} \mathbf{x}_{(h,s)} &= \mathbf{L}_{(h,s)}^{(-1)} \cdot \mathbf{B}_{(h,s)} \\ \mathbf{y}_{s[g]} &= \frac{\tau}{\tau + \mathbf{N}_{s_{[g]}}} \cdot \mathbf{D}_g \cdot \Sigma_g^{(-1)} \cdot \overline{\mathbf{X}}_{s[g]}, \end{aligned} \quad (8)$$

where $\mathbf{D}_g = \frac{\Sigma_g^{1/2}}{\sqrt{\tau}}$, τ is the MAP relevance factor.

To calculate the i^{th} line of $\mathbf{U}_{[g]}$, we use:

$$\mathbf{U}_{[g]}^i = \mathbf{L}\mathbf{U}_g^{-1} \cdot \mathbf{R}\mathbf{U}_g^i, \quad (9)$$

where $\mathbf{R}\mathbf{U}_g^i$ and $\mathbf{L}\mathbf{U}_g$ are defined as follows:

$$\begin{aligned} \mathbf{L}\mathbf{U}_g &= \sum_s \sum_{h \in s} (\mathbf{x}_{(h,s)} \mathbf{x}_{(h,s)}^T) \cdot \mathbf{N}_{(h,s)_{[g]}} \\ \mathbf{R}\mathbf{U}_g^i &= \sum_s \sum_{h \in s} \overline{\mathbf{X}}_{(h,s)_{[g]}}[i] \cdot \mathbf{x}_{(h,s)}. \end{aligned} \quad (10)$$

The implementation in [8] varies in two aspects. In equation 7 an identity matrix is added to $\mathbf{L}_{(h,s)}$, and the calculation of $\mathbf{L}\mathbf{U}_g$ in equation 10 also needs extra statistics, i.e:

$$\begin{aligned} \mathbf{L}_{(h,s)} &= \mathbf{I} + \sum_{g \in \mathbf{UBM}} \mathbf{N}_{(h,s)_{[g]}} \cdot \{\mathbf{U}\}_{[g]}^t \cdot \mathbf{\Sigma}_{[g]}^{-1} \cdot \{\mathbf{U}\}_{[g]} \\ \mathbf{L}\mathbf{U}_g &= \sum_s \sum_{h \in s} (\mathbf{L}_{(h,s)}^{(-1)} + \mathbf{x}_{(h,s)} \mathbf{x}_{(h,s)}^T) \cdot \mathbf{N}_{(h,s)_{[g]}}. \end{aligned} \quad (11)$$

Note that, to be consistent with the training, the estimation of latent variables ($\mathbf{x}_{(h,s)}$ and $\mathbf{y}_s$) on a single utterance should also use MLE to update the parameters $\mathbf{x}_{(h,s)}$ (Equation 8).

Given the factor rank is much less than the number of supervectors (we estimate 40 ranks out of thousands of session examples), and we also have enough data for each session (typically around 2mins after removing the non-speech data), the degeneracy problem stated in [6] resulting from MLE should not happen. This is verified by the experiment in section 5.

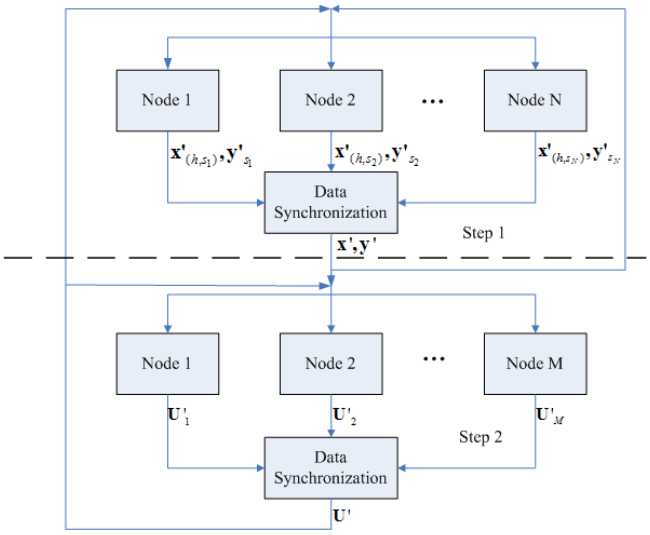

Figure 1: System diagram of the parallelized implementation. N is the number of speakers and M is the number of Gaussians.

4.2. Parallelized Training

As mentioned in the previous section, the parallelization can be performed in two stages: the estimation of each speaker and session dependent parameters and the estimation of transformation matrix $\mathbf{U}_{[g]}$. These two steps are explained as follows. Since the general statistics is estimated only once based on UBM and will be always used in the following procedure, the calculation of $\mathbf{N}_s$, $\mathbf{N}_{(h,s)}$, $\mathbf{X}_s$ and $\mathbf{X}_{(h,s)}$ is treated as a pre-processing step and performed in advance.

Parallelization of speaker-session factors estimation

The estimation of speaker and session dependent parameters is performed speaker by speaker. For each speaker:

- Input the general statistics $\mathbf{N}_s$, $\mathbf{N}_{(h,s)}$, $\mathbf{X}_s$ and $\mathbf{X}_{(h,s)}$, the transformation $\mathbf{U}$, the speaker and channel factor $\mathbf{x}_{(h,s)}$ and $\mathbf{y}_s$ of the latest iteration to calculate the center statistics $\overline{\mathbf{X}}_s$ and $\overline{\mathbf{X}}_{(h,s)}$;
- Estimate the speaker and channel factor $\mathbf{x}'_{(h,s)}$ and $\mathbf{y}'_s$ of this iteration;

Parallelization of transformation matrix U estimation

The estimation of transformation matrix $\mathbf{U}$ is performed mixture by mixture. For each Gaussian mixture (the index of Gaussian mixture is omitted without confusion):

- Input $\mathbf{x}'_{(h,s)}$ and the general statistics $\mathbf{N}_{(h,s)}$ for all speakers to calculate $\mathbf{LU}$;
- Given $\mathbf{y}'_s$ of this iteration, update the center statistics $\overline{\mathbf{X}}'_{(h,s)}$ to calculate $\mathbf{RU}$;
- Update $\mathbf{U}'$ according to $\mathbf{LU}$ and $\mathbf{RU}$.

4.3. Implementation Issues

The work flow of the parallelized implementation is illustrated by figure 1. The general statistics are omitted from the input without confusion.

Although the system performance is quite similar, our implementation has some advantages especially to parallelize the calculation:

- The interface is cleaner and the input/output is easy to manage. In step 1 we input the matrix $\mathbf{U}$ and channel/speaker factors $\mathbf{x}, \mathbf{y}$ of the latest iteration to calculate the channel/speaker factors for each speaker respectively; after that, we input the updated channel/speaker factors and the matrix of the latest iteration to update the session variability matrix of each mixture respectively;
- The storage of interim statistics is minimized. The only statistics we need to save are speaker and channel factors and the variability matrix. If we follow the implementation in [8], the center statistics $\overline{\mathbf{X}}_{(h,s)}$ and $\mathbf{L}_{(h,s)}$ for each speaker should also be saved, which will bring heavy file I/O burden.

5. Experimental Results

All the experiments were carried out on the NIST SRE'05 data which is provided for one-conversation two-channel condition task of the NIST SRE'05 evaluation[2]. The experiments only refer to the NIST defined core condition which includes 23095 trials The primary performance measure is the Detection Cost Function (DCF) defined as a weighted sum of missed detections and false alarms, the normalized cost taking the following form $C_{Norm} = 0.1 \times P_{Miss} + 0.99 \times P_{FalseAlarm}$. In this paper, we report the Minimal DCF (MDC) value obtained a posteriori. The Equal Error Rate (EER) is also provided as another performance measure.

The GMM-UBM system was implemented as in [10]. The front-end feature extraction uses 15 PLP + 15 Delta PLP + 15 Delta-Delta PLP + 1 Delta Energy + 1 Delta-Delta Energy which makes a 47-dimensional vector. The training data was chosen from target speakers in NIST SRE'97-'01 and '03 evaluations and test speakers in NIST SRE'03 evaluation. This data was separated into 3 categories (cell./carb./elec.) and a GMM with 512 components was trained on each set, resulting in a GMM with 1536 Gaussians after fusion.

The rank number (R) of factor analysis model was fixed to be 40, trained on SRE'04 data. To perform score normalizations (t-norm/z-norm/zt-norm) [11], 250 male session and 250 female session data were chosen randomly from SRE'04. Gender-dependent score normalization was performed.

5.1. Experiments on Feature Normalization

In this experiment different front-end feature normalization techniques including CMS, feature mapping and feature warping are compared in the context of factor analysis. The results are given in table 1.

The trials were carried out in both the forward direction (that is, test utterance vs. target model designations given by NIST) and in the reverse direction. The two strategies give similar results but averaging the scores gives additional improvements.

It was found that it is necessary to perform another iteration feature normalization after subtracting session variability for each frame based on equation 2. That is because the distribution of features is distorted after the factor analysis.

According to the experimental results, factor analysis is quite robust and can be used with different feature normalization techniques. The improvements are consistently around 30%∼40% in EER and 15%∼20% in MDC relatively (under the same condition of using t-norm, forward only scoring).

[2]http://www.nist.gov/speech/tests/sre/2005/index.html.

	Conf.	baseline	no Norm			t-norm			z-norm			zt-norm		
			F.	B.	F+B	F.	B.	F+B	F.	B.	F+B	F.	B.	F+B
EER (%)	*S1*	22.69	21.83	24.12	21.00	15.80	18.17	13.38	13.10	15.67	**10.48**	11.60	13.01	10.60
	S2	13.22	19.79	21.53	18.25	13.01	12.56	**9.56**	14.34	15.14	12.27	11.10	11.85	9.82
	S3	10.49	9.85	11.39	8.82	6.15	6.90	**5.36**	6.57	8.23	5.82	5.95	6.70	5.49
	S4	9.48	10.52	11.02	8.73	6.74	7.53	**5.54**	6.57	8.27	5.82	5.94	6.91	5.65
MDC $\times 100$	*S1*	6.75	7.20	7.52	6.41	5.33	5.68	4.41	5.54	5.81	4.46	4.27	4.81	**3.85**
	S2	4.56	5.70	5.95	4.82	4.86	4.83	3.87	5.30	5.68	4.43	4.04	4.35	**3.50**
	S3	3.89	4.34	4.15	3.56	3.03	2.81	**2.31**	2.88	3.43	2.58	2.34	2.71	2.33
	S4	3.57	4.03	4.20	3.30	3.08	3.08	2.46	2.87	3.43	2.40	2.53	2.72	**2.36**

Table 1: Performance of different feature normalizations combined with FA on SRE'05. F. stands for forward, B. stands for backward and F+B means averaging the scores from two directions. "S1" stands for the system without feature normalization, "S2" stands for CMS, "S3" stands for feature warping and "S4" stands for feature warping+feature mapping. The baseline number is obtained with t-norm and without factor analysis using forward-only scoring.

The only exception is CMS, which gives better EER but worse MDC. This is probably because we normalize the speech after removing non-speech data and merge all the data into one segment. It was also found that the feature mapping is no longer necessary to get good performance. Although the baseline using feature mapping and feature warping are slightly better than feature warping only, the best performance is obtained with feature warping after factor analysis. The score normalization is also necessary to get good performance.

5.2. Experiments on Parallelized Maximum Likelihood Estimation

The parallelized fast training was applied on a cluster server with IBM bi-proc blades using OSCAR (open source cluster application resources) [12] system. Feature warping, the best feature normalization configuration according to previous experiments, was used in this experiment.

With only one single node (2.8G Intel Xeon CPU, 2G memory), it took 74.4 hours for the male model training (124 speakers, 1197 sessions) and 138.9 hours for the female model training (186 speakers, 1743 sessions). Convergence was reached after 20 iterations. Running on 20 nodes (2.8G Intel Xeon CPU, 2G memory, training both the male and female model at the same time), the computation time was reduced from several days to several hours. Each iteration was done within 1.5 $\sim$ 2.0 hours. However, similar performance is obtained with MLE compared to LIA's implementation.

	Conf.	no Norm			t-norm		
		F.	B.	F+B	F.	B.	F+B
EER (%)	LIA toolkit	9.85	11.39	8.82	6.15	6.90	5.36
	MLE	9.81	11.35	8.86	6.19	6.87	5.36
MDC $\times 100$	LIA toolkit	4.34	4.15	3.56	3.03	2.81	2.31
	MLE	4.33	4.14	3.55	3.04	2.82	2.31

Table 2: Performance of LIA's implementation and MLE algorithm on SRE'05.

6. Conclusions

The practical issues in using factor analysis model for automatic speaker verification including the comparison of feature normalization techniques and the parallelized implementation of model training based on maximum likelihood estimation are discussed in this paper. We demonstrate that factor analysis can be well accompanied with feature normalization techniques, however feature mapping is no more necessary when FA is performed. The best performance is obtained with feature warping and without feature mapping. The parallelized implementation speeds up the model training without sacrificing the performance.

7. Acknowledgment

The authors would like to thank LIA for making the LIA speaker recognition toolkit available. The implementation of factor analysis in this paper is based on LIA's toolkit.

8. References

[1] B. Fauve, D. Matrouf, N. Scheffer, J.-F. Bonastre, and J. Mason, "State-of-the-art performance in text-independent speaker verification through open-source software," in *IEEE Transactions on Audio, Speech and Language Processing*, vol. (15), 2007, pp. 1960–1968.

[2] S. Furui, "Cepstral analysis technique for automatic speaker verification," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. (29), 1981, pp. 254–272.

[3] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *Proc. ICASSP 2003*, 2003, pp. 53–56.

[4] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Odyssey 2001*, Grete, Greece, 2001, pp. 213–218.

[5] J.-L. Gauvain and C.-H. Lee, "Maximum a-posteriori estimation for multivariate gaussian mixture observations of markov chains," in *IEEE Transactions on Speech and Audio Processing*, vol. (2), 1994, pp. 291–298. citeseer.ist.psu.edu/gauvain94maximum.html

[6] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," in *IEEE Transactions on Speech and Audio Processing*, vol. (13), 2005, pp. 345–354.

[7] R. Vogt, B. Baker, and S. Sridharan, "Modeling session variability in text-independent speaker verification," in *Proc.Interspeech'2005*, Lisboa,Portugal, 2005, pp. 809–812.

[8] D. Matrouf, N. Scheffer, B. Fauve, and J.-F. Bonastre, "A straightforward and efficient implementation of the factor analysis model for speaker verification," in *Proc. Interspeech'2007*, Antwerp, Belgium, 2007, pp. 1242–1245.

[9] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," in *IEEE Transaction on Speech Audio Processing*, 2000, pp. 695–707.

[10] M. Ferras, C.-C. Leung, C. Barras, and J.-L. Gauvain, "Constrained mllr for speaker recognition," in *Proc. ICASSP 2007*, 2007, pp. 53–56.

[11] C. Barras and J.-L. Gauvain, "Feature and score normalization for speaker verification of cellular data," in *Proc. ICASSP 2003*, 2003, pp. 49–52.

[12] B. des Ligneris, S. L. Scott, T. Naughton, and N. Gorsuch, "Open source cluster application resources (oscar): design, implementation and interest for the [computer] scientific community," in *HPCS 2003*, 2003, pp. 241–246.