ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Supervector LDA: A New Approach to Reduced-Complexity I-vector Language Recognition

Alan McCree[1] *and Bengt J. Borgström*

MIT Lincoln Laboratory, Lexington, MA, USA
jonas.borgstrom@ll.mit.edu

10.21437/Interspeech.2012-22

Abstract

In this paper, we extend our previous analysis of Gaussian Mixture Model (GMM) subspace compensation techniques using Gaussian modeling in the supervector space combined with additive channel and observation noise. We show that under the modeling assumptions of a total-variability i-vector system, full Gaussian supervector scoring can also be performed cheaply in the total subspace, and that i-vector scoring can be viewed as an approximation to this. Next, we show that covariance matrix estimation in the i-vector space can be used to generate PCA estimates of supervector covariance matrices needed for Joint Factor Analysis (JFA). Finally, we derive a new technique for reduced-dimension i-vector extraction which we call Supervector LDA (SV-LDA), and demonstrate a 100-dimensional i-vector language recognition system with equivalent performance to a 600-dimensional version at much lower complexity.

Index Terms: language recognition, Gaussian mixture model, Wiener filter, factor analysis, i-vector, LDA.

1. Introduction

Gaussian Mixture Models (GMMs) have long been successful for both speaker and language recognition, and recently subspace methods have been shown to provide convenient models for channel compensation and speaker/language modeling, particularly with the Joint Factor Analysis approach [1]. More recently, even the subspace parameters themselves, referred to as i-vectors, have been used for recognition [2]. State-of-the-art acoustic language recognition has been demonstrated with SVM and Gaussian classifiers in i-vector space in [3, 4].

We recently presented an alternative perspective on GMM subspace algorithms based on sufficient statistics and an additive noise model, and showed that this provides justifications for both JFA and i-vector algorithms [5]. In this paper, we extend this analysis to show that under the i-vector subspace assumptions, full supervector Gaussian scoring can also be performed in the subspace dimension. We also show that for long duration speech cuts, i-vector and supervector Gaussian scoring systems are identical. Using these relationships, we show that supervector hyperparameters can be estimated from i-vector covariance matrices, and explore dimension reduction techniques. We conclude by showing that a new technique, supervector LDA (SV-LDA), allows low-dimension i-vector extraction with equivalent performance at greatly reduced complexity.

2. Statistical Framework

In this section, we introduce the statistical framework used throughout the paper, and discuss efficient subspace scoring techniques based on matrix Wiener filtering.

This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

2.1. Sufficient Statistics Scoring

Evaluating the likelihood for a set of successive frames under a GMM model is traditionally determined as the product of frame-specific likelihoods. However, it is well known that this can equivalently be evaluated by first computing sufficient statistics for the ensemble of frames, and then using a single formula for the total likelihood. This sufficient statistic scoring proves beneficial for computation of GMM likelihoods in the particular instance when the frame alignment is already given, for example from a universal background model (UBM). Although the assumption of UBM-alignment results in a small performance degradation, the use of a single set of sufficient statistics is critical for computational feasibility of sophisticated modeling and scoring techniques. Using sufficient statistics reduces the evaluation of the GMM likelihood for a set of frames to the evaluation of a single Gaussian in the *supervector* space created by stacking GMM mixture means.

2.2. The Additive Noise Model

This leads to the additive noise model [5]:

$$\bar{\mathbf{x}} = \mathbf{m}_i + \mathbf{c} + \mathbf{n}, \tag{1}$$

where $\bar{\mathbf{x}}$ denotes the test sample mean supervector derived from sufficient statistics, $\mathbf{m}_i$ is the supervector mean of language i, $\mathbf{c}$ is additive channel noise, and $\mathbf{n}$ is additive observation noise. Throughout this paper, we define $\bar{\mathbf{x}}$ and $\mathbf{m}_i$ to be normalized with respect to the UBM model mean. The observation noise is Gaussian with zero mean and covariance $\mathbf{\Sigma}_n$ which shrinks to zero as the number of observed frames increases. The observation noise covariance can then be defined in the supervector space as a block diagonal matrix with each diagonal element being the corresponding covariance from the UBM covariance $\mathbf{\Sigma}_0$ divided by the count for this Gaussian. That is,

$$\mathbf{\Sigma}_n = \mathbf{N}^{-1}\mathbf{\Sigma}_0, \tag{2}$$

where $\mathbf{N}$ is a diagonal matrix comprised of diagonal blocks $N_m\mathbf{I}$, $\mathbf{I}$ is the appropriately-sized identity matrix, and N_m is the count for GMM mixture m.

We model $\mathbf{m}_i$ and $\mathbf{c}$ to each be Gaussian with zero mean and covariances $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_c$, respectively. $\mathbf{\Sigma}_s$ is often called the *across-class covariance*, and describes the variability of well-trained language models. $\mathbf{\Sigma}_c$, on the other hand, is often refered to as the *within-class covariance*, and describes the variability of language supervectors about corresponding well-trained models. Using the additive noise model, the likelihood for language i can be expressed as

$$\bar{\mathbf{x}}|S_i \sim \mathcal{N}(\mathbf{m}_i, \mathbf{\Sigma}_n + \mathbf{\Sigma}_c), \tag{3}$$

which we refer to as *full Gaussian scoring* (in JFA terminology this is integration over channel factors). Note that the assumptions of UBM alignment and mean-only models will be made throughout the rest of this paper.

[1] now with the HLTCOE at Johns Hopkins (alan.mccree@jhu.edu)

2.3. Subspace Modeling

In [5], we modeled both within-class and across-class covariances with PPCA. In this paper, motivated by the success of i-vector systems [2], we use the concept of total variability defined by $\mathbf{\Sigma}_t$=$\mathbf{\Sigma}_s + \mathbf{\Sigma}_c$, and make the more restrictive assumption that both channel and language inhabit the same low-dimensional total variability subspace. This allows all covariances to be expressed using PCA

$$\mathbf{\Sigma}_c = \mathbf{U}_c\mathbf{U}_c^T, \mathbf{\Sigma}_s = \mathbf{U}_s\mathbf{U}_s^T, \text{ and } \mathbf{\Sigma}_t = \mathbf{U}_t\mathbf{U}_t^T. \tag{4}$$

Using the previous assumptions, we can write

$$\mathbf{m}_i = \mathbf{U}_t\mathbf{U}_t^+\mathbf{m}_i \tag{5}$$

where the matrix pseudoinverse of $\mathbf{U}_t$ is given by

$$\mathbf{U}_t^+ = \left(\mathbf{U}_t^T\mathbf{U}_t\right)^{-1}\mathbf{U}_t^T. \tag{6}$$

Additionally, we will often utilize the matrix inversion lemma, which can be expressed as [6]

$$\left(\mathbf{D} + \mathbf{UAU}^T\right)^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{U} \times \left(\mathbf{A}^{-1} + \mathbf{U}^T\mathbf{D}^{-1}\mathbf{U}\right)^{-1}\mathbf{U}^T\mathbf{D}^{-1}, \tag{7}$$

and the following related identity

$$\mathbf{U}^T\left(\mathbf{D} + \mathbf{UAU}^T\right)^{-1} = \mathbf{A}^{-1}\left(\mathbf{A}^{-1} + \mathbf{U}^T\mathbf{D}^{-1}\mathbf{U}\right)^{-1}\mathbf{U}^T\mathbf{D}^{-1}. \tag{8}$$

3. Subspace Scoring Methods

In [5], it was shown that under the additive noise model, minimum mean square error (MMSE) noise suppression results in matrix Wiener filtering. Suppressing the channel noise results in the JFA algorithm, while compensating for observation noise results in two different subspace scoring algorithms. In this section we summarize these two i-vector approaches and then show that under our subspace modeling assumptions full Gaussian scoring can also be evaluated in the subspace. Finally we show the similarity of these expressions, particularly for long-duration test cuts.

3.1. Using Wiener Filtering to Derive I-vectors

In this section, we review the use of matrix Wiener filtering to derive model-dependent (MD) and model-independent (MI) i-vectors.

3.1.1. Model Dependent I-vectors

Following (1), the MMSE estimate of an observation noise-compensated supervector can be determined using a model-dependent Wiener filter:

$$\hat{\mathbf{x}}_i^{md} = \mathbf{\Sigma}_c(\mathbf{\Sigma}_c + \mathbf{\Sigma}_n)^{-1}(\bar{\mathbf{x}} - \mathbf{m}_i) + \mathbf{m}_i. \tag{9}$$

The model likelihood can then be evaluated using only channel noise:

$$\hat{\mathbf{x}}_i^{md}|S_i \sim \mathcal{N}(\mathbf{m}_i, \mathbf{\Sigma}_c). \tag{10}$$

Although $\mathbf{\Sigma}_c$ is singular, this log-likelihood can be expressed using the limit

$$\log p\left(\bar{\mathbf{x}}|S_i\right) = \frac{1}{2}\lim_{\delta\to 0}\left(\hat{\mathbf{z}}^{md} - \boldsymbol{\mu}_i^{md}\right)^T \times \mathbf{U}_c^T\left(\delta\mathbf{I} + \mathbf{U}_c\mathbf{U}_c^T\right)^{-1}\mathbf{U}_c\left(\hat{\mathbf{z}}^{md} - \boldsymbol{\mu}_i^{md}\right) + f\left(\bar{\mathbf{x}}\right) \tag{11}$$

where the *model-dependent channel i-vector* is given by:

$$\hat{\mathbf{z}}^{md} = \left(\mathbf{I} + \mathbf{U}_c^T\mathbf{\Sigma}_n^{-1}\mathbf{U}_c\right)^{-1}\mathbf{U}_c^T\mathbf{\Sigma}_n^{-1}\bar{\mathbf{x}}, \tag{12}$$

and

$$\boldsymbol{\mu}_i^{md} = \left(\mathbf{I} + \mathbf{U}_c^T\mathbf{\Sigma}_n^{-1}\mathbf{U}_c\right)^{-1}\mathbf{U}_c^T\mathbf{\Sigma}_n^{-1}\mathbf{m}_i. \tag{13}$$

Also, $f\left(\bar{\mathbf{x}}\right)$ is a function of $\bar{\mathbf{x}}$ which is irrelevant during evaluation of (11) for closed-set identification; such constant terms are ignored in the rest of this paper for the sake of notational brevity. Applying (8) to (11) yields

$$\log p\left(\bar{\mathbf{x}}|S_i\right) = \frac{1}{2}\left(\hat{\mathbf{z}}^{md} - \boldsymbol{\mu}_i^{md}\right)^T\left(\hat{\mathbf{z}}^{md} - \boldsymbol{\mu}_i^{md}\right) \tag{14}$$

Thus, (10) can be evaluated equivalently as

$$\hat{\mathbf{z}}^{md}|S_i \sim \mathcal{N}(\boldsymbol{\mu}_i^{md}, \mathbf{I}), \tag{15}$$

which represents a Gaussian in the channel subspace.

3.1.2. Model Independent I-vectors

As an alternative to Sec. 3.1.1, we can replace (9) with a model-independent observation noise compensation Wiener filter

$$\hat{\mathbf{x}}^{mi} = (\mathbf{\Sigma}_s + \mathbf{\Sigma}_c)(\mathbf{\Sigma}_s + \mathbf{\Sigma}_c + \mathbf{\Sigma}_n)^{-1}\bar{\mathbf{x}}. \tag{16}$$

The model likelihood is then

$$\hat{\mathbf{x}}^{mi}|S_i \sim \mathcal{N}(\mathbf{m}_i, \mathbf{\Sigma}_c). \tag{17}$$

Following steps similar to those above, it can be shown that this is equivalent to

$$\hat{\mathbf{z}}^{mi}|S_i \sim \mathcal{N}\left(\boldsymbol{\mu}_i^{mi}, \mathbf{\Phi}_c\right), \tag{18}$$

where the *total variability i-vector* is given by:

$$\hat{\mathbf{z}}^{mi} = \left(\mathbf{I} + \mathbf{U}_t^T\mathbf{\Sigma}_n^{-1}\mathbf{U}_t\right)^{-1}\mathbf{U}_t^T\mathbf{\Sigma}_n^{-1}\bar{\mathbf{x}}, \tag{19}$$

and the mean in the total variability subspace is given by

$$\boldsymbol{\mu}_i^{mi} = \mathbf{U}_t^+\mathbf{m}_i. \tag{20}$$

Here, $\mathbf{\Phi}_c$ denotes the subspace covariance matrix in the total variability space, so that

$$\mathbf{\Sigma}_c = \mathbf{U}_t\mathbf{\Phi}_c\mathbf{U}_t^T. \tag{21}$$

From this, an analytic form of $\mathbf{\Phi}_c$ can be derived as

$$\mathbf{\Phi}_c = \mathbf{U}_t^+\mathbf{\Sigma}_c\mathbf{U}_t^{+,T} = \mathbf{U}_t^+\mathbf{U}_c\mathbf{U}_c^T\mathbf{U}_t^{+,T}. \tag{22}$$

Note that this total variability scoring is equivalent to the generative model Gaussian scoring in [4].

3.2. Subspace Full Gaussian Scoring

We now show that under these subspace assumptions, full Gaussian supervector scoring can also be evaluated with an efficient subspace-based algorithm. The log-likelihood for (3) is

$$\log p\left(\bar{\mathbf{x}}|S_i\right) = -\frac{1}{2}\left(\bar{\mathbf{x}} - \mathbf{m}_i\right)^T\left(\mathbf{\Sigma}_n + \mathbf{\Sigma}_c\right)^{-1}\left(\bar{\mathbf{x}} - \mathbf{m}_i\right). \tag{23}$$

Applying (4) and (5) yields

$$\log p\left(\bar{\mathbf{x}}|S_i\right) = \tag{24}$$
$$-\frac{1}{2}\left(\mathbf{U}_c\mathbf{U}_c^{+}\mathbf{m}_i\right)^T\left(\boldsymbol{\Sigma}_n+\mathbf{U}_c\mathbf{U}_c^T\right)^{-1}\left(\mathbf{U}_c\mathbf{U}_c^{+}\mathbf{m}_i-2\bar{\mathbf{x}}\right).$$

Using (8) leads to

$$\log p\left(\bar{\mathbf{x}}|S_i\right) = -\frac{1}{2}\left(\mathbf{U}_c^{+}\mathbf{m}_i\right)^T\mathbf{H}_c^{-1}\mathbf{U}_c^{+}\mathbf{m}_i + \left(\mathbf{U}_c^{+}\mathbf{m}_i\right)^T\mathbf{H}_c^{-1}\left(\mathbf{U}_c^T\boldsymbol{\Sigma}_n^{-1}\mathbf{U}_c\right)^{-1}\mathbf{U}_c^T\boldsymbol{\Sigma}_n^{-1}\bar{\mathbf{x}}, \tag{25}$$

where

$$\mathbf{H}_c = \left(\mathbf{U}_c^T\boldsymbol{\Sigma}_n^{-1}\mathbf{U}_c\right)^{-1}\left(\mathbf{I}+\mathbf{U}_c^T\boldsymbol{\Sigma}_n^{-1}\mathbf{U}_c\right). \tag{26}$$

By completing the square, we see that full Gaussian scoring is equivalently evaluated as

$$\mathbf{z}^{gs,c}|S_i \sim \mathcal{N}\left(\boldsymbol{\mu}_i^{gs,c},\mathbf{H}_c\right), \tag{27}$$

where

$$\mathbf{z}^{gs,c} = \left(\mathbf{U}_c^T\boldsymbol{\Sigma}_n^{-1}\mathbf{U}_c\right)^{-1}\mathbf{U}_c^T\boldsymbol{\Sigma}_n^{-1}\bar{\mathbf{x}}, \tag{28}$$

and

$$\boldsymbol{\mu}_i^{gs,c} = \mathbf{U}_c^{+}\mathbf{m}_i. \tag{29}$$

In this way, full Gaussian scoring is implemented as the evaluation of a single Gaussian in the low-dimensional channel subspace. Note that these formulas are similar in form but not in detail to the model-dependent channel i-vector equations in Sec. 3.1.1.

Alternatively, using

$$\mathbf{U}_c = \mathbf{U}_t\mathbf{U}_t^{+}\mathbf{U}_c, \tag{30}$$

full Gaussian scoring can be expressed as

$$\mathbf{z}^{gs,t}|S_i \sim \mathcal{N}\left(\boldsymbol{\mu}_i^{gs,t},\mathbf{H}_t\right), \tag{31}$$

where

$$\mathbf{z}^{gs,t} = \left(\mathbf{U}_t^T\boldsymbol{\Sigma}_n^{-1}\mathbf{U}_t\right)^{-1}\mathbf{U}_t^T\boldsymbol{\Sigma}_n^{-1}\bar{\mathbf{x}}, \tag{32}$$

the subspace mean is given by

$$\boldsymbol{\mu}_i^{gs,t} = \mathbf{U}_t^{+}\mathbf{m}_i, \tag{33}$$

and

$$\mathbf{H}_t = \mathbf{U}_t^{+}\mathbf{U}_c\mathbf{H}_c\mathbf{U}_c^T\mathbf{U}_t^{+,T}. \tag{34}$$

Here, full Gaussian scoring shows great similarity to total variability i-vectors presented in Sec. 3.1.2.

3.3. Long-Duration Approximations

It is of interest to study the behavior of full Gaussian scoring in the case of long-duration cuts. Particularly, it is useful to compare full Gaussian scoring with model-dependent channel and total variability i-vectors. By considering (2), it can be concluded that for long duration signals, (26) can be approximated by $\mathbf{H}_c \approx \mathbf{I}$. This is due to the fact that for large T, the effect of the identity matrix in the right-most term of (26) becomes negligible. Using similar reasoning, (34) can be shown to be approximated by $\mathbf{H}_t \approx \boldsymbol{\Phi}_c$ for large T. For the case of long-duration cuts, full Gaussian scoring can then be approximated by evaluating (27) with $\mathbf{H}_c$=$\mathbf{I}$, or (31) with $\mathbf{H}_t$=$\boldsymbol{\Phi}_c$. These correspond to the long-duration approximations of model-dependent channel and total variability i-vectors, respectively. Therefore, all three subspace methods are equivalent for large T.

From (26), $\mathbf{H}_c$ can be interpreted as applying attenuation to sufficient statistics based on signal duration. Specifically, for short signals, $\mathbf{H}_c$ attenuates sufficient statistics to normalize for observation noise. Conversely, for long signals, $\mathbf{H}_c$ nears the identity matrix, and has no effect on scoring. Note that this is similar in concept to duration modeling proposed in [7]. By contrast, the duration behavior of i-vector scoring methods is less intuitive, since for example the total variability scoring of (18) compares an attenuated i-vector with a non-attenuated mean.

4. Subspace Hyperparameter Estimation

The previous sections show the explicit relationships between covariance matrices estimated in the total variability subspace and their full supervector counterparts. Here, we show that this can be used as a hyperparameter estimation technique, and show how to correctly apply PCA for dimension reduction in the full supervector space. Finally, we derive a new technique, which we call Supervector LDA (SV-LDA), for reducing the dimension of the total variability space used for i-vector extraction.

4.1. Subspace Estimation

Estimating within-class and across-class covariance matrices in the total variability space is straightforward, since i-vectors are of low dimension. Also, since the Wiener filtering has removed the observation noise from the sufficient statistics, only language and channel variations remain. Equation (21) shows the relationship between the subspace covariance matrix $\boldsymbol{\Phi}_c$ and the full supervector $\boldsymbol{\Sigma}_c$ covariance, and implies an algorithm to estimate $\mathbf{U}_c$ from $\boldsymbol{\Phi}_c$. After eigendecomposition of a subspace sample covariance matrix $\boldsymbol{\Phi}_c$, the supervector matrix $\mathbf{U}_c$ is attained by pre-multiplying by $\mathbf{U}_t$. This approach can be applied for $\mathbf{U}_s$ as well; it can even be used to produce an updated version of $\mathbf{U}_t$ itself with *iterative Wiener filtering*.

4.2. Dimension Reduction

It is common in JFA to use lower dimensions for $\mathbf{U}_s$ and $\mathbf{U}_c$. While it is tempting to use PCA to reduce the dimensions of the subspace covariance matrices and then map them back to the full supervector space, this is not effective since it does not account for the eigenvalues within $\mathbf{U}_t$. The correct way to accomplish this, for example for $\mathbf{U}_c$, is to first use $\boldsymbol{\Phi}_c$ to generate the supervector matrix $\mathbf{U}_c$, and then to perform PCA dimension reduction on $\mathbf{U}_c\mathbf{U}_c^T$. For an i-vector system, this supervector PCA (SV-PCA) approach provides a simple way to reduce the dimension of $\mathbf{U}_t$ without retraining.

4.3. Supervector LDA

LDA is often used in i-vector systems to reduce the dimensionality of scoring [2]. However, this is done after full-sized i-vector extraction, so it does not reduce the complexity of the entire system. Motivated by the previous section, we now show how to extend LDA back to the original supervector space, so that the total variability subspace defined by the matrix $\mathbf{U}_t$ can be designed to optimize discrimination.

Using LDA, the subspace $\mathbf{U}_t$ can be constructed as the highest-energy eigenvectors of $\boldsymbol{\Sigma}_c^{-1}\boldsymbol{\Sigma}_s$. Performing the eigendecomposition directly in the full supervector space is prohibitively complex. However, by applying the transpose of (8),

Scoring Method	Test Duration		
	30s	10s	3s
I-vec Gaussian	2.2	4.9	14.5
Supervector Gaussian	2.1	5.0	16.7

Table 1: Min C_{avg} performance as a function of scoring method for NIST LRE09 Evaluation Set.

and taking the limit of a PPCA inverse:

$$\begin{aligned}\boldsymbol{\Sigma}_c^{-1}\boldsymbol{\Sigma}_s &= \lim_{\delta\to 0}\left(\delta\mathbf{I}+\mathbf{U}_t\boldsymbol{\Phi}_c\mathbf{U}_t^T\right)^{-1}\mathbf{U}_t\boldsymbol{\Phi}_s\mathbf{U}_t^T \\ &= \lim_{\delta\to 0}\mathbf{U}_t\left(\delta\boldsymbol{\Phi}_c^{-1}+\mathbf{U}_t^T\mathbf{U}_t\right)^{-1}\boldsymbol{\Phi}_c^{-1}\boldsymbol{\Phi}_s\mathbf{U}_t^T \\ &= \mathbf{U}_t\left(\mathbf{U}_t^T\mathbf{U}_t\right)^{-1}\boldsymbol{\Phi}_c^{-1}\boldsymbol{\Phi}_s\mathbf{U}_t^T.\end{aligned} \tag{35}$$

Here, $\boldsymbol{\Phi}_s$ is the language model covariance matrix in the total variability subspace, and is defined similarly to (22). Using the kernel trick, the eigendecomposition required for LDA can therefore be performed with the decomposition of a subspace matrix. In practice, regularizing each subspace sample covariance matrix to a diagonal version allows both $\boldsymbol{\Phi}_c$ and $\boldsymbol{\Phi}_s$ to be of full rank so that any number of LDA dimensions can be generated, not just the number of language classes. In addition, we have empirically found better performance by forcing this subspace matrix to be symmetric using only the upper triangular portion.

5. Experimental Results

This section presents experimental comparisons of these systems for the NIST LRE 2009 evaluation set.

5.1. System Design

The feature extraction, training data, and back-end used in this work are similar to that employed in [8]. Speech is windowed at 20 ms with a 10 ms frame rate, filtered through a mel-scale filter bank followed by RASTA. Each vector is then converted into a 56-dimensional shifted delta cepstra vector using a 7-1-3-7 scheme and concatenation to the static cepstral coefficients. Speech activity detection is then applied and the speech is normalized to a standard distribution.

The data used for training and development includes two main sources of data, conversational telephone speech (CTS) and broadcast news (BN). The CTS partition includes data from multiple corpora including CallFriend, CallHome, Mixer, OHSU and the OGI-22 collections. The BN partition includes data from VoA as supplied by NIST. The evaluation data used is the data defined by NIST for the 2009 LRE and includes evaluation segments for 30s, 10s and 3s covering 23 language classes.

For the i-vector system, a 2048-component UBM is used, with a 600-dimensional i-vector extractor trained on the same data from these 23 languages. The back-end uses a discriminatively trained Gaussian with shared covariance. In these experiments, duration norming [7] is not applied since the i-vector systems have a natural downweighting of short duration scores.

5.2. Results

Results for the total variability i-vector system on LRE09 are given in Tables 1 and 2. The first table shows that subspace full Gaussian scoring provides equivalent performance to i-vector scoring, although it is somewhat worse for short durations. Note that overall this performance is comparable to [3] and not quite as good as [4]; we believe this is due to the lack of vocal tract normalization. Table 2 shows that dimension reduction in the full i-vector space via LDA provides equivalent performance, while reducing the complexity of scoring but not i-vector extraction. The remaining rows show that SV-PCA provides reduced complexity extraction, but with significant degradation even at 200 dimensions, while SV-LDA degrades much less. In fact, the 100-dimensional SV-LDA system provides equivalent performance to the baseline while reducing the amount of CPU time needed for each i-vector extraction from 3 seconds to 0.1.

Reduction Method	Dimension		
	50	100	200
LDA	2.2/5.0/14.7	2.2/5.1/14.7	2.2/5.1/14.7
SV-PCA	5.2/8.6/18.4	3.1/6.1/15.8	2.6/5.2/14.8
SV-LDA	2.5/5.3/14.8	2.3/5.1/14.6	2.2/5.0/14.4

Table 2: Min C_{avg} for 30/10/3 second testing as a function of dimension for NIST LRE09 Evaluation Set.

6. Conclusion

In this paper, we have extended our previous analysis of GMM subspace techniques to highlight the similarities between JFA and i-vector systems. Experiments on NIST LRE09 have confirmed that both methods can share hyperparameter estimation and provide similar performance. In addition, we proposed a new techique called SV-LDA that uses a discriminative criterion to reduce the dimension and complexity of i-vector extraction without degrading performance.

7. Acknowledgements

The authors gratefully acknowledge interesting discussions related to this work with colleagues Doug Reynolds, Doug Sturim, and Fred Richardson.

8. References

[1] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, pp. 980–988, July 2008.

[2] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, pp. 788–798, May 2011.

[3] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, 2011, pp. 857–860.

[4] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space," in *Proc. Interspeech*, 2011, pp. 861–864.

[5] A. McCree, D. Sturim, and D. Reynolds, "A new perspective on GMM subspace compensation based on PPCA and Wiener filtering," in *Proc. Interspeech*, 2011, pp. 145–148.

[6] A. Laub, *Matrix Analysis for Scientists and Engineers*, SIAM, 2004.

[7] A. McCree, F. Richardson, E. Singer, and D. Reynolds, "Beyond frame independence: Parametric modeling of time duration in speaker and language recognition," in *Proc. Interspeech*, 2008, pp. 767–770.

[8] P. Torres-Carrasquillo, E. Singer, T. Gleason, A. McCree, D. Reynolds, F. Richardson, and D. Sturim, "The MIT/LL NIST LRE 2009 language recognition system," in *ICASSP*, 2010, pp. 4994–4997.