JFA-BASED FRONT ENDS FOR SPEAKER RECOGNITION

Patrick Kenny, Themos Stafylakis, Pierre Ouellet and Md. Jahangir Alam

Centre de recherche informatique de Montréal (CRIM)

{Patrick.Kenny, Themos.Stafylakis, Pierre.Ouellet, Jahangir.Alam}@crim.ca

ABSTRACT

We discuss the limitations of the i-vector representation of speech segments in speaker recognition and explain how Joint Factor Analysis (JFA) can serve as an alternative feature extractor in a variety of ways. Building on the work of Zhao and Dong, we implemented a variational Bayes treatment of JFA which accommodates adaptation of universal background models (UBMs) in a natural way. This allows us to experiment with several types of features for speaker recognition: speaker factors and diagonal factors in addition to i-vectors, extracted with and without UBM adaptation in each case. We found that, in text-independent speaker verification experiments on NIST data, extracting i-vectors with UBM adaptation led to a 10% reduction in equal error rates although performance did not improve consistently over the whole DET curve. We achieved a further 10% reduction (with a similar inconsistency) by using speaker factors extracted with UBM adaptation as features. In text-dependent speaker recognition experiments on RSR2015 data, we were able to achieve very good performance using a JFA model with diagonal factors but no speaker factors as a feature extractor. Contrary to standard practice, this JFA model was configured so as to model speaker-phrase combinations (rather than speakers) and it was trained on utterances of very short duration (rather than whole recording sessions). We also present a variant of the length normalization trick inspired by uncertainty propagation which leads to substantial gains in performance over the whole DET curve.

Index Terms— speaker recognition, joint factor analysis, i-vectors, PLDA, variational Bayes

1. INTRODUCTION

The i-vector/PLDA approach to text-independent speaker recognition [1, 2] is now so well established that the limitations of the i-vector representation of speech segments have started to become apparent. The sensitivity of i-vectors to segment durations is an obvious case in point but this can be dealt at the PLDA level [3]; other shortcomings are not so easily handled in the back end and this has led us to investigate some new ways of using the hidden variables in Joint Factor Analysis (JFA) modeling as features for speaker recognition. Thus we are planning to explore the use of JFA as a "front end" in the sense of [1] rather than as a classifier in its own right.

Recall that if the recordings of a speaker are indexed by r, the general JFA model assumes that recording r is represented by a supervector of the form

$$\boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_r + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{D}\boldsymbol{z} \tag{1}$$

(using standard notation). Here $\boldsymbol{x}_r$ is a low-dimensional recording-dependent vector of channel factors, $\boldsymbol{y}$ is a low-dimensional vector of speaker factors and $\boldsymbol{z}$ is a vector of diagonal factors of supervector size. Like the $\boldsymbol{y}$-vectors, the $\boldsymbol{z}$-vectors are recording-indepdent and hence can be used to characterize speakers [4, 5].

I-vectors arose by suppressing the terms $\boldsymbol{y}$ and $\boldsymbol{z}$, so that all recordings are treated as being statistically independent (regardless of the association between recordings and speakers). Since no distinction is made between speaker and channel variation, the i-vector variant of JFA is particularly easy to implement. The fact that channel and speaker effects are commingled in the i-vector features appears at first sight to be a disadvantage but it turns out that a PLDA classifier is well equipped to disentangle these effects. A key ingredient of the success of PLDA in text-independent speaker recognition is that i-vectors can be taken to be *low dimensional*: in JFA terminology, both speaker and channel variation can be well accounted for by relatively small numbers of speaker and channel factors.

However the sensitivity of i-vectors to channel effects causes problems in situations where there are severe channel degradations. This became apparent in the NIST 2012 speaker recognition evaluation where many of the trials involved speech degraded by additive noise. Standard multi-condition training would dictate that speech corrupted by additive noise should also be used in training i-vector extractors but it turned out that using only clean speech was the better strategy. Thus i-vector extractors trained on noisy speech do not produce features which are well suited to distinguishing between speakers.

This suggests a first experiment with alternative features: train a JFA model of the form $\boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_r + \boldsymbol{V}\boldsymbol{y}$ and use the $\boldsymbol{y}$-vectors rather than the i-vectors as features. Like i-vectors, the $\boldsymbol{y}$ vectors can be taken to be low dimensional but, unlike i-vectors, they ought to be robust to severe channel distortions (provided that such distortions are adequately represented in the JFA training set). This is apparent from the formula for the MAP estimate of $\boldsymbol{y}$, namely

$$\langle \boldsymbol{y} \rangle = \boldsymbol{P}^{-1} \sum_r \sum_c \boldsymbol{V}_c^* \boldsymbol{\Sigma}_c^{-1} (\boldsymbol{F}_c^r - N_c^r \boldsymbol{m}_c - N^r \boldsymbol{U}_c \langle \boldsymbol{x}^r \rangle)$$

where $\boldsymbol{P}$ is the posterior precision of $\boldsymbol{y}$. Here N_c^r and $\boldsymbol{F}_c^r$ are the zero and first order statistics associated with mixture component c in recording r and $\boldsymbol{\Sigma}_c$ is the covariance matrix of the mixture component. The term $\boldsymbol{U}_c \langle \boldsymbol{x}^r \rangle$ serves to correct the first order Baum-Welch statistics by removing the channel effects in a manner which is similar to the way Baum-Welch statistics are collected from noisy speech using the vector Taylor series (VTS) method [6].

Likewise, one can use $\boldsymbol{z}$-vectors rather than $\boldsymbol{y}$-vectors as features. These supervector size features are not as easy to model as speaker factors so they will not perform as well as $\boldsymbol{y}$-vector features but there is evidence that low dimensional subspaces cannot capture all useful information about speaker identities [7]. Information encoded in the $\boldsymbol{z}$-vectors is particularly useful in cases where there are multiple enrollment recordings for a speaker [8] and, even in the case of a single enrollment recording, $\boldsymbol{z}$-vectors contain information which is complementary to $\boldsymbol{y}$-vectors [2].

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

That speaker and channel variability in text-independent speaker recognition can be treated as being mostly low dimensional is now well established, at least if enrollment and test utterances are of long duration (as in the NIST speaker recognition evaluations). However in text-dependent speaker recognition, the objects to be recognized are speaker-phrase combinations rather than speakers as such and speaker-phrase combinations do not appear to be low dimensional at all. (Phonetic variability dominates speaker and channel variability in short utterances and phoneme strings do not admit low dimensional descriptions). I-vector extractors trained on text-independent data do not translate well to text dependent task domains and it seems that i-vector methods in text-dependent speaker recognition can only be made to perform well if speakers are constrained to use a universal passphrase and data is collected from large numbers of speakers for each deployment [9, 10] (and papers cited there).

These considerations suggest that a JFA model of the form $\boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}^r + \boldsymbol{D}\boldsymbol{z}$ might yield better features than i-vectors for text-dependent speaker recognition, provided that the model is implemented in such a way that the $\boldsymbol{z}$-vectors characterize speaker-phrase combinations rather than speakers. Note that (as in traditional GMM/UBM modeling), $\boldsymbol{z}$-vectors extracted from utterances of short duration can be expected to be sensitive to their phonetic content; this would be a drawback in text-independent speaker recognition but not in the text-dependent situation where it is always the case that the same phrase is encountered at enrollment and test time. Moreover, if only a limited amount of development data is available to build a text-dependent system, then using $\boldsymbol{z}$-features would make it possible to build a lightweight system (comparable to relevance MAP/NAP).

We will report the results of text-independent speaker recognition experiments performed using $\boldsymbol{y}$ and $\boldsymbol{z}$ features on NIST data and of text-dependent speaker recognition experiments performed using $\boldsymbol{z}$-features on RSR2015 data [11] using PLDA-like and cosine distance based back end classifiers. We will also investigate another question which arises generally in extracting features of this sort (including i-vectors), namely whether Baum-Welch statistics ought to be collected with a universal background model (as is generally done) or whether the universal background model (UBM) ought to be adapted to the data first (as is done in the case of VTS modeling, for instance, [6]). The JFA model (1) was originally intended as a way of adapting GMM supervectors to data so adapting the UBM would seem to be the most natural course. Early work in subspace methods used this type of adaptation [12, 13, 14] and it is important in subspace GMM modeling for speech recognition [15] but very little has been published on this question in the context of JFA or i-vector modeling. (The only exceptions that we are aware of are [16] and [6].) One would expect that UBM adaptation should be effective in any situation where there is a gross mismatch between the data and the UBM; examples might be additive noise at low SNRs or lexical mismatch in a text-dependent speaker recognition application where each user has her own password.

2. ALGORITHMS

The key computation in extracting features (be they i-vectors, $\boldsymbol{y}$-vectors or $\boldsymbol{z}$-vectors) from a collection of one or more recordings is to calculate the joint posterior distribution of the hidden variables in the JFA model. It is well known that this calculation is straightforward in the i-vector case, provided that there is no UBM adaptation (so that Baum-Welch statistics are collected in the usual way) [13]. Under the same conditions, the joint posterior in a general JFA model can be calculated by brute force [4] or, much more efficiently, by the Gauss-Seidel method used in [17]. Although it is not presented as such, the Gauss-Seidel method is an instance of variational Bayes and it provides a very good approximation to the joint posterior in that it is guaranteed to find the mode of the posterior exactly if it is allowed to run to convergence. In the most general situation where UBM adaptation is allowed and the assignments of frames to Gaussian mixture components need to be treated as hidden variables on the same footing as the continuous hidden variables in (1), some approximate posterior computation such as variational Bayes is unavoidable. (This is the case even for i-vectors.) The variational Bayes posterior calculations for a general JFA model with UBM adaptation have been worked out in [16] and we use these for extracting features in our experiments.

In [16] the authors experimented with UBM adaption in a JFA classifier for text-independent speaker verification. The role of variational Bayes was to supply evidence lower bounds for use in the JFA likelihood ratio calculations. Although UBM adaptation was carried out at run time (that is, enrollment and testing) the JFA model was trained without adapting the UBM to the recordings in the training corpus. (Training was performed using the methods in [5].) In order to avoid this inconsistency, we implemented a variational Bayes EM training algorithm for JFA which is applicable with or without UBM adaption.

Turning now to the back end, let us focus on $\boldsymbol{y}$-features to fix ideas. Note that if we are given multiple recordings for a speaker, these will enable us to extract a *single* $\boldsymbol{y}$-vector rather than multiple feature vectors as in the i-vector case. In the NIST context there are typically about 10 i-vectors per speaker available for PLDA training. (Although i-vector averaging may be used to collapse multiple i-vectors to a single i-vector at run time, i-vector averaging is not used in PLDA training.) On the other hand, in order to train a PLDA classifier for a y-vector system, we need to fabricate a training set consisting of pairs of "enrollment" and "test" y-vectors by partitioning the recordings of the training speakers in various ways. A $\boldsymbol{y}$-vector extracted from a large number of recordings ought to be treated as being more reliable than one extracted from a small number of recordings and uncertainty propagation provides a mechanism to incorporate this type of information into PLDA [3]. However we did not use uncertainty propagation in the experiments reported here as we found that it did not perform satisfactorily. (On the other hand, we will present a simple modification to the length normalization trick which is inspired by the uncertainty propagation idea and which turns out to be very effective.)

As for the $\boldsymbol{z}$ features, they can be modeled in a similar way provided that the PLDA matrices are constrained to be diagonal. This constraint is necessary since the $\boldsymbol{z}$ features are of supervector size but it has the unfortunate consequence that the number of free parameters to estimated is relatively modest so that the resulting classifiers are not very powerful. (Simple cosine distance based classifiers perform almost as well.)

3. TEXT-INDEPENDENT EXPERIMENTS

We used data provided in the NIST speaker recognition evaluations up to and including 2010 to construct a UBM and i-vector training set comprising 49K utterances from 3134 speakers. We reserved 100 female speakers for testing and we created one model per speaker. (Thus the test set was disjoint from the training set, contrary to the set up in the 2012 evaluation.) The number of target and nontarget trials was 1312 and 166914 respectively. The number of enrollment utterances per speaker varied randomly between 1 and 6 and no distinction was made between telephone and microphone utterances, ei-

ther in devising trial lists or in selecting enrollment utterances. (We took care to avoid using parallel microphone recordings and same-number telephone recordings.)

We used a standard front end (60 dimensional MFCC features with short term Gaussianization) and a 512 component UBM with diagonal covariance matrices.

3.1. i-vectors

For our experiments with i-vectors on the NIST data we trained i-vector extractors of dimension 400 with and without UBM adaptation. The results presented in Table 1 indicate that UBM adaptation yields about a 10% relative improvement in equal error rate (EER) but that there is not a uniform improvement over the whole DET curve (NDCF refers to the NIST normalized detection cost functions as defined in the 2008 and 2010 evaluation plans). We were concerned that there might be a danger of overfitting in adapting the UBM to each recording in our training and test sets. To prevent this happening we scaled the zero and first order Baum-Welch statistics by a factor which is denoted by θ in the Table. The results show that no scaling at all ($\theta = 1$) actually works just as well as scaling by a factor of 1/3 so we need not have bothered.

Table 1. 400 dimensional i-vectors, with and without UBM adaptation

	EER	2008 NDCF	2010 NDCF
no adaptation	1.95%	**0.047**	0.28
adaptation ($\theta = 1$)	**1.8**%	0.054	0.26
adaptation ($\theta = 1/3$)	**1.8**%	0.053	0.26

Note on i-vector averaging

For all of the experiments reported in Table 1 we used "by the book" scoring, that is, we used the PLDA model to calculate likelihood ratios for speaker verification in the way prescribed by the rules for manipulating probabilities. Participants in the 2012 NIST speaker recognition found that it was necessary to use "i-vector averaging" instead, that is to create a single i-vector for each speaker at enrollment time by averaging the i-vectors extracted from each of the speaker's enrollment recordings. The results in Table 2 show that i-vector averaging is clearly deleterious in the case of our test set and this explains why we did not use it in our experiments. Note that in

Table 2. By the book scoring vs. i-vector averaging

	EER	2008 NDCF	2010 NDCF
by the book	**1.95**%	**0.047**	**0.28**
i-vector averaging	2.26%	0.067	0.33

the NIST 2012 test set there were speakers with very large numbers of enrollment recordings (up to 100 including many instances where recordings were made with multiple microphones and so cannot be considered to be statistically independent) whereas in our test the number of enrollment recordings for each speaker was kept within reasonable bounds (between 1 and 6). Thus the posterior distribution of the identity variable which characterizes the speaker in by the book PLDA modeling can never become excessively sharp (as would to happen if by the book scoring is applied under the conditions of the 2012 NIST evaluation).

3.2. y-features

We trained JFA models with 400 speaker factors and 200 channel factors with and without UBM adaptation (so that the dimension of the $\boldsymbol{y}$-vectors is 400). As in our i-vector experiments, we observed a large increase in supervector variances with UBM adaptation, with most of the excess variance being on the channel side rather than the speaker side. As the back-end, we used the PLDA variant described in Section 2. The results in Table 3 show that UBM adaptation leads to a further 10% relative improvement in EER (compared with the best result in Table 1) but again we see that there is not a uniform improvement in performance over the whole DET curve.

Table 3. 400 dimensional $\boldsymbol{y}$-vectors, JFA trained with and without UBM adaptation

	EER	2008 NDCF	2010 NDCF
no adaptation	1.76%	0.057	0.34
adaptation ($\theta = 1/3$)	1.56%	0.069	0.41

Note on length normalization

It is well known that length normalization of i-vectors is crucial to the success of Gaussian PLDA modeling. Since we used PLDA classifiers for our experiments with the $\boldsymbol{x}$ and $\boldsymbol{y}$ features, we used length normalization in all cases but we implemented the variant introduced in [18]. The standard procedure for producing a length normalized $\boldsymbol{y}$-vector would be to calculate the posterior expectation $\langle \boldsymbol{y} \rangle$ and divide this by its length $\|\langle \boldsymbol{y} \rangle\|$; a more sophisticated approach would be to estimate the length of the $\boldsymbol{y}$-vector in a way which takes account of the posterior covariance matrix $\mathrm{Cov}\,(\boldsymbol{y}, \boldsymbol{y})$, so that the length is estimated as the square root of $\|\langle \boldsymbol{y} \rangle\|^2 + \mathrm{tr}\,(\mathrm{Cov}\,(\boldsymbol{y}, \boldsymbol{y}))$. This is the variant of length normalization that we used in our experiments with the $\boldsymbol{y}$ features reported in Table 3. If we use traditional length normalization instead, we get the results reported in Table 4. It is evident that traditional length normalization leads to uniform degradations in performance across all operating points.

Table 4. As in Table 3 but with traditional length normalization

	EER	2008 NDCF	2010 NDCF
no adaptation	1.96%	0.069	0.38
adaptation ($\theta = 1/3$)	1.98%	0.079	0.44

3.3. z-features

Table 5 presents some results on the NIST data obtained using a JFA model of the form $\boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_r + \boldsymbol{D}\boldsymbol{z}$ (with $\boldsymbol{U}$ of rank 200) as a feature extractor and a diagonal PLDA classifier as described in Section 2. A classical way of estimating $\boldsymbol{D}$ is to use relevance MAP: for a given relevance factor r, $\boldsymbol{D}$ is chosen so that $r\boldsymbol{D}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{D} = \boldsymbol{I}$ [4]. The term "ML II" refers to maximum likelihood II estimation (this is the criterion used in the variational Bayes EM training algorithm for JFA). It is apparent from the table that relevance MAP gives a better EER than ML II but that performance elsewhere on the DET curve is similar. Unfortunately, UBM adaptation was unsuccessful.

Table 5. JFA with 200 channel factors and diagonal term, with and without UBM adaptation, r = relevance factor

	EER	2008 NDCF	2010 NDCF
r = 8, no adaptation	2.8%	0.106	0.47
ML II, no adaptation	3.33%	0.108	0.46
ML II, adaptation	4.92%	0.158	0.55

4. TEXT-DEPENDENT EXPERIMENTS

For these experiments we used the Part I portion of RSR2015 dataset. We used the same front end and UBM configuration as for our text independent experiments but the UBM as well as the JFA models that we tested were trained on the Part I background data. This consists of parallel recordings of 30 TIMIT phrases uttered by 47 female and 50 male speakers, each of whom participated in 9 recording sessions. Contrary to [18, 19] we did not use the development data for training but we used the same test set as in those papers.

Each target speaker model was created using three recordings of a phrase and the same phrase was used for all verification trials involving the speaker model. (The lexical content varied from one trial to another so one of the challenges is to set decision thresholds in a phrase-independent way.) The total number of speaker models was about 3K and there were about 18K target trials and 1M nontarget trials.

The JFA models that we trained were of the form $\boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_r + \boldsymbol{D}\boldsymbol{z}$. The simplest possibility is not to do UBM adaptation and to estimate $\boldsymbol{D}$ by relevance MAP; in the case of text-dependent speaker recognition a typical relevance factor would be 4. The hidden variable $\boldsymbol{z}$ serves to characterize speaker-phrase combinations, not speakers. A crude benchmark can be obtained by suppressing the $\boldsymbol{x}_r$ term. Table 6 reports the results for female trials obtained with a cosine distance classifier and the PLDA classifier described in Section 2 In this instance the PLDA classifier outperforms the

Table 6. JFA with no channel factors and diagonal term, no UBM adaptation, r = relevance factor

	EER	2008 NDCF	2010 NDCF
cosine	3.40%	0.15	0.62
PLDA	3.07%	0.13	0.57

cosine distance classifier but that is not surprising since the cosine distance classifier has no way of coping with session variability in this case (as the term $\boldsymbol{x}_r$ has been suppressed). For our other (more realistic) experiments, the performance gap is much less. Tables 7 and 8 show that good results can be obtained if the term $\boldsymbol{x}_r$ is restored and standard score normalization techniques are applied in conjunction with cosine distance scoring. These results are similar to the best results reported in [11, 10, 18, 19]. The interesting thing about them is that they were obtained by a completely new method. Training a JFA model so as to model speaker-phrase combinations rather than speakers using utterances of 2–4 seconds duration rather than whole recording sessions is not something that has previously been attempted

We tried several other experiments which yielded results which were worse than those in Table 6 (the crude benchmark) so we will not describe them in detail. Thus we trained a JFA model on whole recording sessions rather than 2–4 second utterances and got poor results (as expected). Furthermore we were unable to get improvements either from maximum likelihood II estimation of $\boldsymbol{D}$ or from UBM adaption. These results were not what we hoped for but they are consistent with our experience with the $\boldsymbol{z}$-features in the text-independent case (Section 3.3).

Table 7. JFA with 50 channel factors and diagonal term, without UBM adaptation, trained on 2 second utterances. relevance factor 4. results on female speakers.

	EER	2008 NDCF	2010 NDCF
no normalization	2.04%	0.094	0.47
t-norm	1.46%	0.067	0.51
z-norm	1.54%	0.075	0.50
s-norm	1.29%	0.059	0.44

Table 8. As in Table 7 male trials rather than female

	EER	2008 NDCF	2010 NDCF
no normalization	1.38%	0.072	0.28
t-norm	0.93%	0.049	0.20
z-norm	1.52%	0.074	0.28
s-norm	1.04%	0.053	0.20

5. DISCUSSION

We have presented some encouraging results obtained by using JFA models as feature extractors for speaker recognition. This approach was motivated by two considerations: the limitations of the i-vector representation of speech segments (particularly for text-dependent speaker recognition with arbitrary passphrases) and the success of the i-vector/PLDA cascade which can be viewed as one way of decomposing JFA into front-end and back-end models. This success is largely due to the fact that it enables the length normalization trick to be applied, something that cannot be accommodated in a monolithic JFA classifier. Roughly speaking, applying length normalization to high dimensional data has the effect of Gaussianizing it so the benefit of deployng JFA as a feature extractor rather than as a monolithic classifier is that it enables us to overcome the limitations of Gaussian modeling in JFA.

This paper breaks new ground in applying JFA to the problem of text-dependent speaker verification. Although they are quite preliminary, the results that we have reported are very encouraging. We study this topic in much greater detail in [20].

Acknowledgements

Most of this work was done at the speaker recognition workshop held at Johns Hopkins University in July 2013. Thanks to the Center for Language and Speech Processing for their hospitality.

6. REFERENCES

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[2] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Proc. Odyssey 2010: The speaker and Language Recognition Workshop*, Brno, Czech Rebublic, June 2010.

[3] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for Speaker Verification with Utterances of Arbitrary Duration ," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[4] P. Kenny, "Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms, Tech. Report CRIM-06/08-13," 2005. [Online]. Available: http://www.crim.ca/perso/patrick.kenny

[5] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 16, no. 5, pp. 980–988, July 2008. [Online]. Available: http://www.crim.ca/perso/patrick.kenny

[6] Y. Lei, L. Burget, and N. Scheffer, "A noise robust i-vector extractor using Vector Taylor Series for speaker recognition," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[7] H. Aronowitz, "Speaker recognition using Kernel-PCA and inter-session variability modeling," in *Proc. Interspeech 2007*, Antwerp, Belgium, Aug. 2007, pp. 298–301.

[8] P. Kenny, N. Dehak, R. Dehak, V. Gupta, and P. Dumouchel, "The role of speaker factors in the NIST extended data task," in *Proc. IEEE Odyssey Workshop*, Stellenbosch, South Africa, Jan. 2008.

[9] H. Aronowitz and O. Barkan, "On leveraging conversational data for building a text dependent speaker verification system," in *Proc. Interspeech*, Lyon, France, Sept. 2013.

[10] A. Larcher, K.-A. Lee, B. Ma, and H. Li, "Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[11] ——, "The RSR2015 database for text-dependent speaker verification using multiple pass-phrases," in *Proc. Interspeech*, Portland OR, Sept. 2012.

[12] P. Kenny, M. Mihoubi, and P. Dumouchel, "New MAP estimators for speaker recognition," in *Proc. Eurospeech*, Geneva, Switzerland, Sept. 2003, pp. 2691–2694. [Online]. Available: http://www.crim.ca/perso/patrick.kenny

[13] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. Speech Audio Processing*, vol. 13, no. 3, pp. 345–359, May 2005.

[14] N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. van Leeuwen, P. Matejka, P. Schwarz and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 7, pp. 2072 – 2084, Sept. 2007.

[15] D. Povey, L. Burget, M. Agarwal, *et al.*, "The subspace Gaussian mixture model – a structured model for speech recognition," *Computer Speech and Language*, 2011.

[16] X. Zhao and Y. Dong, "Variational Bayesian Joint Factor Analysis Models for Speaker Verification," *IEEE Trans. Audio Speech and Language Processing*, vol. 20, no. 3, pp. 1032–1042, 2012.

[17] R. Vogt, B. Baker, and S. Sridharan, "Modeling session variability in text-independent speaker verification," in *Proc. Eurospeech*, Lisbon, Portugal, Sept. 2005, pp. 3117–3120.

[18] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, M. Kockmann, and P. Dumouchel, "I-Vector/PLDA Variants for Text-Dependent Speaker Recognition," *in preparation*.

[19] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, and M. Kockmann, "Text-dependent speaker recognition using PLDA with uncertainty propagation," in *Proc. Interspeech*, Lyon, France, Sept. 2013.

[20] P. Kenny, T. Stafylakis, J. Alam, P. Ouellet, and M. Kockmann, "Joint Factor Analysis for Text-Dependent Speaker Verification," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Joensuu, Finland, June 2014.