The Speaker and Language Recognition Workshop (Odyssey 2016)
21-24 June 2016, Bilbao, Spain

Uncertainty Modeling Without Subspace Methods For Text-Dependent Speaker Recognition

Patrick Kenny,[1] *Themos Stafylakis,*[1] *Jahangir Alam,*[1] *Vishwa Gupta,*[1] *and Marcel Kockmann*[2]

[1]CRIM, Canada, {patrick.kenny, themos.stafylakis, vishwa.gupta, jahangir.alam}@crim.ca
[2]VoiceTrust, Germany, marcel.kockmann@voicetrust.com

Abstract

We present an effective, practical solution to the problem of uncertainty modeling in text-dependent speaker recognition where "uncertainty" refers to the fact that feature vectors used for speaker recognition are necessarily noisy in the statistical sense if they are extracted from utterances of short duration. The idea is to apply the I-Vector Backend probability model at the level of individual Gaussian mixture components rather than at the supervector level. We show that (unlike the I-Vector Backend), this approach can be implemented in a way which makes reasonable computational demands at verification time. Uncertainty modeling enables us to achieve error rate reductions of up to 25% on the RSR Part III speaker verification task (compared to an implementation of the Joint Density Backend [8] which treats point estimates of supervector features as being reliable).

1. Introduction

This paper is concerned with backend modeling in text-dependent speaker recognition where channel-compensated Baum-Welch statistics are used as the feature representation. Considering that large universal background models (e.g. 512 Gaussians) prove to be effective in text-dependent speaker recognition even though test utterances are typically short (e.g. 2 seconds), these Baum-Welch statistics are very sparse so that point estimates of the features which are usually used to characterize speakers in text-dependent speaker recognition need to be treated as uncertain.

How to deal with this uncertainty is an open problem in speaker recognition generally but some progress has been made in the text-independent situation by characterizing the uncertainty in point estimates of i-vectors in a PLDA backend by means of posterior covariance matrices specified by zero order Baum-Welch statistics [1, 2]. The evidence in favour of this approach is not overwhelmingly convincing (it has only been shown to work well under conditions of extreme mismatch) and because it depends critically on characterizing speakers by subspace methods (i-vectors or speaker factors), it has not had much impact on the problem domain where it would seem to be most acutely needed, namely text-dependent speaker recognition with very short utterances.

Except in rare situations where large amounts of background data happen to be available, best results in text-dependent speaker recognition are obtained with supervector-sized features rather than subspace methods. PLDA cannot generally serve as a backend classifier, but the Joint Density Backend [3, 4, 8] can play the role of a trainable backend for text-dependent speaker recognition in its stead. The Joint Density Backend is based on a model of the joint distribution of enrollment and test feature vectors in a speaker verification trial under the same speaker-hypothesis. It can be configured to work with either supervector or subspace features. Although we will occasionally refer to the subspace version of the Joint Density Backend in order to explain how the ideas presented in this paper build on our earlier work, we will be concerned solely with the problem of extending the capability of the supervector version of the Joint Density Backend to accommodate uncertainty modeling in the present paper. We will show that on the RSR Part III (random digits) development and evaluation test sets [5], this uncertainty modeling results in error rate reductions of 20% or more compared to the Joint Density Backend.

Although subspace methods are of limited use in text-dependent speaker recognition, in studying the uncertainty modeling problem it is natural to start with them considering that some progress has already been made in this way in the text-independent case. Because the results we obtained on text-dependent tasks with the uncertainty propagation version of i-vector/PLDA were unsatisfactory [6], we developed a more fundamental approach to the problem in [7]. Rather than treat the enrollment and test feature vectors in a target trial as being statistically independent for the purpose of uncertainty modeling (an obviously erroneous assumption made in [1, 2]), we showed in that paper how to quantify the uncertainty in the point estimates of such features in a *trial dependent way*, so that the uncertainty in a test utterance diminishes the more enrollment data is available. When this uncertainty is accommodated in (the subspace version of) the Joint Density Backend, it turns out that the likelihood ratio calculations for speaker verification

 10.21437/Odyssey.2016-3

are formally equivalent to evidence calculations with i-vector extractors having non-standard normal priors so we dubbed this approach the "I-Vector Backend". In this paper, we develop a similar approach to accommodating uncertainty in the supervector version of the Joint Density Backend, leading to the "Hidden Supervector Backend". (Appendix A in this paper shows how the I-Vector Backend and the Hidden Supervector Backend can be treated in a unified way.)

It turns out that, as in the subspace case, it is easier to estimate robustly the joint distribution of hidden variables which account for enrollment and test feature vectors than it is to estimate the joint distribution of point estimates of these feature vectors (as in the original formulation of the Joint Density Backend). A particular consequence of this is that, in the RSR Part III application domain, we can make the parameters of the Hidden Supervector Backend digit-dependent, as in the I-Vector Backend [7] (but contrary to both the subspace or supervector versions of the Joint Density Backend).

In [8] we described another method we developed to introduce digit-dependency into the supervector version of the Joint Density Backend which we refer to as component fusion. (The idea is to weight the contributions of individual mixture components to the speaker verification likelihood ratio in a digit-dependent way. This method only works with supervector features, not subspace features.) We will also show in this paper that this component fusion method carries over successfully to the Hidden Supervector Backend.

We note that we have previously performed an extensive study of the Joint Density Backend on the RSR Part III dataset [8]. The results obtained in this paper with the Hidden Supervector Backend can be directly compared with the results reported there and they turn out to be uniformly better.

2. Background

This paper builds on a series of earlier contributions [7, 4, 8] which we summarize in this section.

2.1. JFA for Speaker Verification with Digits

We model speakers' pronunciation of digits with a tied-mixture HMM (one set of mixture weights for each digit) combined with a JFA model of digit supervectors of the form

$$\boldsymbol{m} + \boldsymbol{D}\boldsymbol{z}(d) + \boldsymbol{U}\boldsymbol{x}^r. \tag{1}$$

Here d is used to indicate a generic digit. Using a tied mixture HMM (in which a single set of "mixture components" is shared among the digits) enables channel effects to be modeled by an utterance level hidden variable $\boldsymbol{x}^r$ (r for recording); $\boldsymbol{z}(d)$ characterizes a speaker's pronunciation of the digit and serves as a supervector sized feature for speaker recognition. The matrix $\boldsymbol{D}$ is estimated by relevance MAP [9].

An alternative configuration ignores the left-to-right structure of enrollment and test utterances altogether. If Baum-Welch statistics are collected with a conventional UBM (or by forced alignment with a speech recognition system as in Appendix A) we can model utterance supervectors as

$$\boldsymbol{m} + \boldsymbol{D}\boldsymbol{z} + \boldsymbol{U}\boldsymbol{x}^r \tag{2}$$

where the hidden variable $\boldsymbol{z}$ is digit-independent. In this case, enrollment or test data is characterized by a single "global" $\boldsymbol{z}$-vector. Following the terminology in [10, 8] we refer to the digit-dependent vectors $\boldsymbol{z}(d)$ in (1) as "local" $\boldsymbol{z}$-vectors. We will restrict ourselves to local $\boldsymbol{z}$-vectors in the expository portion of this paper but we will report the results of speaker verification experiments with both local and global $\boldsymbol{z}$-vectors. As in [8], local and global feature vectors fuse well at the score level.

2.2. The Joint Density Backend (Supervector Version)

At enrollment time, each speaker utters the ten digits in random order several times. Because $\boldsymbol{z}$-vectors are tied across all utterances by a speaker, the enrollment process results in one $\boldsymbol{z}$-vector per digit (regardless of the number of recordings available for enrollment). We denote these $\boldsymbol{z}$-vectors by $\boldsymbol{z}_e(d)$ (e for enrollment and d for digit); likewise $\boldsymbol{z}_t(d)$ indicates a $\boldsymbol{z}$ vector extracted from a test utterance.

The supervector version of the Joint Density Backend [4, 8] forms a likelihood ratio for speaker verification of the form

$$\prod_d \frac{P_T(\boldsymbol{z}_e(d), \boldsymbol{z}_t(d))}{P_N(\boldsymbol{z}_e(d), \boldsymbol{z}_t(d))} \tag{3}$$

where d ranges over digits in the test utterance, P_T refers to the joint distribution of feature vector pairs occurring in target trials and P_N to the joint distribution in non-target trials. We assume that the denominators in (3) factorize as $P_T(\boldsymbol{z}_e(d))P_T(\boldsymbol{z}_t(d))$. Furthermore, we decompose each factor in (3) as a product of terms of the form

$$\frac{P_T(\boldsymbol{z}_{e,c}(d), \boldsymbol{z}_{t,c}(d))}{P_T(\boldsymbol{z}_{e,c}(d))P_T(\boldsymbol{z}_{t,c}(d))} \text{ or } \frac{P_T(\boldsymbol{z}_{t,c}(d)|\boldsymbol{z}_{e,c}(d))}{P_T(\boldsymbol{z}_{t,c}(d))} \tag{4}$$

where, for each mixture component c, $\boldsymbol{z}_{e,c}(d)$ is the corresponding subvector of $\boldsymbol{z}_e(d)$ and similarly for $\boldsymbol{z}_{t,c}(d)$.

2.2.1. Semi-Diagonal Constraints

We model the joint distribution of $\boldsymbol{z}_{e,c}(d)$ and $\boldsymbol{z}_{t,c}(d)$ under the same-speaker hypothesis as a multivariate Gaussian of dimension $2F$ where F is the dimension of the acoustic feature vectors. It is not possible in practice to estimate full covariance matrices for this purpose so we treat the individual components of the acoustic feature vectors as being statistically independent and model the $2F \times 2F$ dimensional covariance matrices with diagonal blocks of dimension $F \times F$. We use the term *semi-diagonal* to refer to this structure.

2.2.2. *Component Fusion*

Clearly, it would be desirable to make the parameters of the Joint Density Backend digit-dependent as the mixture components which are best able to discriminate between speakers are likely different for different digits. Estimating covariance matrices in a digit-dependent way does not work in our experience, even if semi-diagonal constraints are imposed. (For a given digit many of the mixture components will be visited rarely and the corresponding covariance matrices cannot be estimated satisfactorily.)

On the other hand, we have found an effective way to introduce digit-dependency into the Joint Density Backend by weighting the mixture component dependent factors in (4) in a digit-dependent way instead. Specifically, we use regularized logistic regression to estimate a set of C weights for each digit (where C is the number of mixture components in the tied mixture codebook) [8].

2.3. I-Vector Backend

Our first clearly successful effort at uncertainty modeling in text-dependent speaker recognition [7] used speaker factors rather than $\boldsymbol{z}$-vectors as features. In place of (1) we assumed a JFA model of the form

$$\boldsymbol{m} + \boldsymbol{V}\boldsymbol{y}(d) + \boldsymbol{U}\boldsymbol{x}^r \tag{5}$$

where the speaker factor vector $\boldsymbol{y}(d)$ is assumed to be of low dimension. Denoting its dimension by R, as a probability model for the digits in a verification trial under the same-speaker hypothesis we used an i-vector extractor of dimension $2CF \times 2R$ whose total variability matrix is defined by

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{V} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V} \end{pmatrix}. \tag{6}$$

The rationale here is that for each digit d involved in a verification trial, there is a hidden variable $\boldsymbol{w}_e(d)$ (namely the speaker factor vector denoted by $\boldsymbol{y}(d)$ in (5)) which represents the target speaker's pronunciation of the digit and a corresponding hidden variable $\boldsymbol{w}_t(d)$ for the speaker in the test utterance. The corresponding CF dimensional supervectors are $\boldsymbol{m} + \boldsymbol{V}\boldsymbol{w}_e(d)$ and $\boldsymbol{m} + \boldsymbol{V}\boldsymbol{w}_t(d)$ so we can represent the concatenation of the two supervectors as

$$\begin{pmatrix} \boldsymbol{m} \\ \boldsymbol{m} \end{pmatrix} + \begin{pmatrix} \boldsymbol{V} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V} \end{pmatrix} \begin{pmatrix} \boldsymbol{w}_e(d) \\ \boldsymbol{w}_t(d) \end{pmatrix}$$

which has the form of an i-vector extractor if we define the total variability matrix as in (6) and we take the "i-vector" $\boldsymbol{w}$ to be the concatenation of $\boldsymbol{w}_e(d)$ and $\boldsymbol{w}_t(d)$. This i-vector model for supervectors of dimension $2CF$ is only used to perform probability calculations with the evidence formula given in Proposition 2 in Appendix B (and not to extract features for speaker recognition).

2.3.1. *Priors*

To apply the evidence formula, we have to supply a prior on $\boldsymbol{w}$. In conventional i-vector modeling there is nothing to be gained by using non-standard priors but for our purposes we need two non-standard normal priors, $P_T(\boldsymbol{w})$ and $P_N(\boldsymbol{w})$, to model the supervectors of dimension $2CF$ in target and non-target trials respectively.

We estimate $P_T(\boldsymbol{w})$ iteratively using the minimum divergence updates described in Proposition 3 in Appendix B), after arranging the JFA training set into a collection of target trials which simulates the development and evaluation test sets. For each trial, we obtain a set of Baum-Welch statistics of dimension $2CF$ by collecting one set from the enrollment data and another from the test data and concatenating the two. The prior $P_N(\boldsymbol{w})$ is obtained by zeroing out the cross correlations in the covariance matrix which defines P_T (that is, by treating the enrollment and test utterances as being statistically independent).

2.3.2. *Predictive Likelihood Ratio*

To perform speaker verification, for a given trial we could concatenate the Baum-Welch statistics extracted from the enrollment and test data (in the same way as we did in estimating $P_T(\boldsymbol{w})$) and form a likelihood ratio for speaker verification by evaluating the evidence twice, once with $P_T(\boldsymbol{w})$ and once with $P_N(\boldsymbol{w})$. However this approach would be computationally extravagant as it involves evidence calculations with an i-vector extractor of rank $2R$.

In fact the computation can be carried out efficiently with the $CF \times R$ dimensional i-vector extractor defined by setting $\boldsymbol{T} = \boldsymbol{V}$ by taking advantage of the fact $\boldsymbol{w}$ decomposes into two parts, one of which accounts for the enrollment data and the other for the test data. (It is important to note that this decomposition would not be possible but for the block diagonal structure of $\boldsymbol{T}$ in (6).) The idea is to calculate a *predictive* likelihood ratio of the form $P(T|E)/P(T)$ where E stands for the enrollment data and T the test data, rather than a joint likelihood ratio of the form $P(E,T)/P(E)P(T)$.

Fix a digit d and a target speaker. To calculate the numerator distribution $P(\cdot|E)$, take the first order Baum-Welch statistics for the digit from the speaker's enrollment data and pad them with 0's to obtain a full set of statistics of dimension $2CF$, and similarly for the zero order statistics. Using the i-vector extractor of dimension

$2CF \times 2R$ and the prior $P_T(\boldsymbol{w})$, calculate the posterior distribution on the hidden variable $\boldsymbol{w}$ as in Proposition 1 in Appendix B). Interpreting $\boldsymbol{w}$ as the concatenation of the enrollment hidden variable $\boldsymbol{w}_e(d)$ and the test utterance hidden variable $\boldsymbol{w}_t(d)$, we obtain a distribution on $\boldsymbol{w}_t(d)$ by marginalization. Performing evidence calculations with the i-vector extractor of dimension $CF \times R$ and this distribution as the prior enables us to evaluate $P(T|E)$ for trials involving the given target speaker. As for $P(T)$, we simply marginalize $P_N(\boldsymbol{w})$ to obtain another, speaker-independent, prior on $\boldsymbol{w}_t(d)$ and calculate the evidence again.

2.3.3. Digit Dependency

We can also take advantage of the block diagonal structure in (6) in another way, namely to impose semi-diagonal constraints on the covariance matrix that specifies $P_T(\boldsymbol{w})$. We showed in [7] how this enabled us to make the priors $P_T(\boldsymbol{w})$ and $P_N(\boldsymbol{w})$ digit-dependent. This strategy is unsuccessful with the Joint Density Backend but it works very well with the I-Vector Backend.

3. Hidden Supervector Backend

We now apply the idea underlying the I-Vector Backend at the level of individual mixture components rather than at the level of supervectors. This leads us to derive a version of the Joint Density Backend described in Section 2.2 which works with supervectors which are hidden rather than observable (whence the "Hidden Supervector Backend").

For a digit d and mixture component c, let $\boldsymbol{O}_{e,c}(d)$ be the collection of acoustic observation vectors for the digit and mixture component in the enrollment data and let $\boldsymbol{O}_{t,c}(d)$ be the collection of acoustic observation vectors in the test data. In a manner analogous to (4), we will calculate a likelihood ratio for speaker verification by combining terms of the form

$$\frac{P(\boldsymbol{O}_{t,c}(d)|\boldsymbol{O}_{e,c}(d))}{P(\boldsymbol{O}_{t,c}(d))} \tag{7}$$

and evaluate these predictive likelihood ratios in a way which is formally identical to the evidence calculations with the I-Vector Backend described in the Section 2.3.2.

Fix a mixture component c. We create two copies of the mixture component which we label by e and t (one to model enrollment data and the other to model test data). For a given speaker verification trial, we concatenate the zero and first order channel-compensated Baum-Welch statistics N_e and $\boldsymbol{F}_e$ (extracted from the enrollment data $\boldsymbol{O}_{e,c}(d)$) and N_t and $\boldsymbol{F}_t$ (extracted from the test data $\boldsymbol{O}_{t,c}(d)$). We model these concatenated Baum-Welch statistics with an "i-vector extractor" of dimension $2F \times 2R$ subject to block diagonal constraints analogous to (6). Note that this Lilliputian "i-vector extractor" together with the prior under the same-speaker hypothesis, $P_T(w)$, vary from one mixture component to another and have to be estimated accordingly.

As in the I-Vector Backend, the block diagonal structure (6) enables us to decompose the i-vector $\boldsymbol{w}$ into two parts, $\boldsymbol{w}_e$ and $\boldsymbol{w}_t$, one of which accounts for the enrollment data and the other for the test data. Hence a predictive likelihood ratio of the form (7) can be calculated efficiently just as in the case of the I-Vector Backend (Section 2.3.2), assuming that the prior $P_T(\boldsymbol{w})$ has been trained by minimum divergence estimation. Moreover, this structure enables us to impose semi-diagonal constraints on the covariance matrix which defines the prior, as in the case of the I-Vector Backend (Section 2.3.3), although some extra clarification is needed for this.

3.1. Digit Dependency

Note that the minimum divergence estimation formulas which we use to estimate the prior $P_T(\boldsymbol{w})$ have to be applied for each mixture component separately. In making the prior digit-dependent as well we are going to encounter very sparse Baum-Welch statistics so it is not obvious that, even with semi-diagonal constraints, a covariance matrix can be trained for each mixture component/digit pair. (In fact, this doesn't work at all in the case of the supervector version of the Joint Density Backend.)

Fortunately the minimum divergence updates in Proposition 3 behave sensibly in situations where training data is scarce. Indeed, it is easily verified that in the extreme case were there are *no* acoustic observations for a given mixture component, the initial estimate of the prior is a fixed point for the re-estimation procedure. It follows that, contrary to our experience with the Joint Density Backend but similar to our experience in [7], there is no difficulty in making the prior digit-dependent. (For each mixture component, we first train the prior in a digit-independent way and use this to initialize digit-dependent training.)

When we come to describe our experiments in Section 4, we will report results obtained with digit-dependent versions of both the Hidden Supervector Backend and the Joint Density Backend (supervector version). In the case of the Joint Density Backend, digit-dependency has a different meaning — it refers to component fusion scheme discussed in Section 2.2.2. But we will also show that component fusion can be successfully accommodated by the Hidden Supervector Backend. Thus two types of digit-dependency are available to the Hidden Supervector Backend and it turns out that they combine well with each other.

3.2. Computational Efficiency

Although the matrices $\boldsymbol{V}$ that specify the mixture component dependent "i-vector extractors" (6) can be estimated in the usual way using the maximum likelihood II principle, our experience has been that this leads to minimal improvements at substantial computational cost. So for practical purposes it seems to be best to take $R = F$ and $\boldsymbol{V}$ to be the identity matrix for each mixture component. (In other words, the prior $P_T(\boldsymbol{w})$ is made to bear the full burden of modeling same-speaker trials.) In this case, Baum-Welch statistics for the given mixture component can be regarded as summarizing noisy observations of a hidden F-dimensional feature vector. Concatenating these feature vectors gives a hidden supervector — these are the "hidden supervectors" referred to in the introduction.

The posterior calculations in Proposition 1 in Appendix B simplify considerably in the case of identity matrices. Recall that we need to perform this posterior calculation in two situations. Firstly, at enrollment time (and in the course of training the prior) we need to perform a posterior calculation using a $2F \times 2F$ dimensional prior precision matrix $\boldsymbol{P}$ that satisfies semi-diagonal constraints. Let $\boldsymbol{F}$ be the $2F \times 1$ vector obtained by concatenating $\boldsymbol{F}_e$ and $\boldsymbol{F}_t$. Let $\boldsymbol{N}$ be the $2F \times 2F$ block diagonal matrix whose diagonal blocks are $N_e\boldsymbol{I}$ and $N_t\boldsymbol{I}$. Then (using the notation of Proposition 1) the posterior covariance and expectation of $\boldsymbol{w}$ are given by

$$\begin{aligned} \boldsymbol{C} &= (\boldsymbol{P} + \boldsymbol{N})^{-1} \\ \langle \boldsymbol{w} \rangle &= \boldsymbol{C}\,(\boldsymbol{P}\boldsymbol{\mu} + \boldsymbol{F})\,. \end{aligned} \tag{8}$$

Note that semi-diagonal constraints on the precision matrix $\boldsymbol{P}$ are inherited by $\boldsymbol{P} + \boldsymbol{N}$ and hence by $\boldsymbol{C}$.

It follows from this that the linear algebra needed for posterior calculation performed at verification time involves only *strictly diagonal* matrices of dimension $F \times F$. (See the description of the predictive likelihood ratio calculation in in Section 2.3.2.) Thus the computational cost of the Hidden Supervector Backend is essentially the same as that of the Joint Density Backend.

3.3. Channel Compensation

Up to now we have abstracted channel effects. In enrolling a speaker, we create a set of synthetic Baum-Welch statistics for each digit d by taking the Baum-Welch statistics in each recording, removing the channel effects and pooling over the enrollment recordings (and similarly for a test utterance). If the Baum-Welch statistics for recording r and mixture component c and digit d are denoted by $N_c^r(d)$ and $\boldsymbol{F}_c^r(d)$ then the synthetic zero and first order statistics are

$$\begin{aligned} N_c(d) &= \sum_r N_c^r(d) \\ \boldsymbol{F}_c(d) &= \sum_r (\boldsymbol{F}_c^r(d) - N_c^r(d)\boldsymbol{U}_c \langle \boldsymbol{x}^r \rangle) \end{aligned} \tag{9}$$

where $\langle \boldsymbol{x}^r \rangle$ is a point-estimate of the hidden variable $\boldsymbol{x}^r$ in (1). These channel-compensated Baum-Welch statistics serve as the "features" for the Hidden Supervector Backend.

3.4. Normalizing the Baum-Welch Statistics

It is well known that inserting a non-linear length normalization step between a JFA-based feature extractor and a Gaussian backend such as PLDA or the Joint Density Backend effectively compensates for the unsatisfactory Gaussian assumptions on which JFA is based. The question arises as to whether a similar type of normalization should be performed on the synthetic Baum-Welch statistics before presenting them to the I-Vector Backend. The answer is yes, and it proves to be very effective.

In the JFA model, the posterior covariance and expectation of the hidden variable $\boldsymbol{z}_c(d)$, $\boldsymbol{K}_c(d)$ and $\langle \boldsymbol{z}_c(d) \rangle$, are given in terms of the corresponding synthetic Baum-Welch statistics by

$$\begin{aligned} \boldsymbol{K}_c(d) &= (\boldsymbol{I} + N_c(d)\boldsymbol{D}_c^*\boldsymbol{D}_c)^{-1} \\ \langle \boldsymbol{z}_c^d(d) \rangle &= \boldsymbol{K}_c(d)\boldsymbol{D}_c^*\boldsymbol{F}_c(d) \end{aligned}$$

for each mixture component c. These expressions enable us to evaluate $\langle \|\boldsymbol{z}(d)\|^2 \rangle$ since

$$\langle \|\boldsymbol{z}_c(d)\|^2 \rangle = \| \langle \boldsymbol{z}_c(d) \rangle \|^2 + \operatorname{tr}(\boldsymbol{K}_c(d))\,. \tag{10}$$

Because the $\boldsymbol{z}$-vectors are supposed to have a standard normal distribution, $\frac{1}{CF}\|\boldsymbol{z}(d)\|^2$ ought to be equal to 1 on average. This suggests that the first order synthetic statistics should be scaled so as to ensure that, for each spoken digit, $\frac{1}{CF}\langle \|\boldsymbol{z}(d)\|^2 \rangle = 1$. (We leave the zero order statistics unchanged.) An experiment reported in Appendix A shows the utility of including the contribution of the posterior covariance matrices in (10).

4. Experiments

These experiments were conducted with a standard, 60-dimensional PLP front end on the RSR2015 Part III development and evaluation test sets. Utterances of less than 1 second duration or SNR of less than 15 dB were rejected so that not all trials were performed [8].

To train the Hidden Supervector and Joint Density Backends, we devised a backend training set by organizing the RSR2015 Part III background digit data into a set of 14 K target trials intended to simulate the trials in the development test set. We used the same data (organized in the conventional way) for JFA training.

For the Joint Density Backend, we present results obtained with both "local" and "global" $\boldsymbol{z}$-vectors as explained in Section 2.1 and [10]. Global $\boldsymbol{z}$-vectors are interesting to work with because their performance can be measured against a simple GMM/UBM benchmark whose performance on this task turns out to surprisingly good. On the other hand, the digit-dependent versions of the Joint Density and Hidden Supervector Backends can only be explored if local $\boldsymbol{z}$-vectors are used.

		norm.?	EER (M/F)	DCF (M/F)
1	GMM	-	4.8%/8.0%	0.217/0.356
2	JDB	-	4.8%/7.6%	0.219/0.353
3	HSB	$\times$	4.5%/6.8%	0.201/0.338
4	HSB	✓	**3.9%/6.1%**	**0.177/0.307**

Table 1: Results on the development set obtained with 128 Gaussians. The systems are a GMM/UBM system, the Joint Density Backend (JDB) and the Hidden Supervector Backend (HSB) both with global z-vectors. Baum-Welch statistics normalization is indicated by "norm".

The Hidden Supervector Backend is presented with Baum-Welch statistics extracted and normalized with a JFA model rather than with feature vectors, so in this case the terms local and global have to be understood as referring to Baum-Welch statistics. It is a convenient abuse of language to refer to local and global z-vectors in this case as well (this can be justified if the terminology is understood to refer to the configuration of the underlying JFA model).

All results presented were obtained with s-norm score normalization.

4.1. Results on the development set (128 Gaussians)

The results of our first experiment are summarized in Table 1. We compared the GMM/UBM benchmark (line 1) with three global z-vector backends, namely the the Joint Density Backend (line 2) and two versions of the Hidden Supervector Backend, one with unnormalized Baum-Welch statistics (line 3) and the other with normalized Baum-Welch statistics (line 4). The metrics are the equal error rate and the detection cost function defined in the 2008 NIST speaker recognition evaluation plan. The Hidden Supervector Backend achieves large error rate reductions with about half of the improvement being accounted for by the procedure for normalizing the Baum-Welch statistics discussed in Section 3.4. We used a relevance factor of 2 both for MAP adaptation (in the case of the GMM/UBM benchmark) and to train the JFA model which we used for the Joint Density Backend and two the Hidden Supervector Backends. We replicated the experiment in line 4 of Table 1 with different relevance factors in order to explore the effect of the relevance factor in normalizing the Baum-Welch statistics (as described in Section 3.4). The results are shown in Table 2 where it is apparent that the normalization procedure is fairly insensitive to the relevance factor (and a relevance factor of 2 turns out to have been a good choice).

Table 3 contains results on the development set obtained with local z-vectors and 128 Gaussians. Making the Joint Density and Hidden Supervector Backends digit-dependent results in major improvements (although,

r	EER (M/F)	DCF (M/F)
0.5	4.3%/6.6%	0.193/0.329
1	4.1%/6.2%	0.179/0.310
2	**3.9%/6.1%**	**0.177/0.307**
4	3.9%/6.3%	0.183/0.307

Table 2: The effect of varying the relevance factor r in JFA training on the Hidden Supervector Backend

		norm.?	digit-dep.?	EER (M/F)	DCF (M/F)
1	JDB	-	$\times$	5.7%/8.3%	0.248/0.397
2	JDB	-	✓	4.4%/5.6%	0.201/0.309
3	HSB	$\times$	$\times$	5.5%/7.8%	0.247/0.392
4	HSB	$\times$	✓	4.9%/6.7%	0.215/0.343
5	HSB	✓	$\times$	5.7%/7.9%	0.244/0.398
6	HSB	✓	✓	**4.2%/5.4%**	**0.184/0.279**

Table 3: Results on the development set obtained with local z-vectors and 128 Gaussians highlighting the benefit of digit-dependency in the backends

as explained in Sections 2.2 and 14, different mechanisms are used to achieve digit dependency in the two cases). Normalizing the Baum-Welch statistics before presenting them to the Hidden Supervector Backend is again seen to result in substantial improvements.

4.2. Results on the development set (512 Gaussians)

The results we obtained on the development set using 512 Gaussians with global and local z-vectors are presented in Tables 4 (global z-vectors) and 5 (local z-vectors). Baum-Welch statistics were normalized in the case of the Hidden Supervector Backend and, in the case of local z-vectors, both the Joint Density Backend and the Hidden Supervector Backend were made digit-dependent. The highlighted result in the last line of the table refers to a system that exploits both types of digit dependency available to the Hidden Supervector Backend, namely digit dependent priors and the component fusion method.

Note that the results obtained with 512 Gaussians are uniformly better than those obtained with 128 Gaussians despite the fact that the test utterances are of short duration. This phenomenon motivated us to explore uncertainty modeling.

4.3. Results on the evaluation set

Finally we report results obtained by score level fusion of local and global hidden supervector systems with 512 Gaussians on both the evaluation and development sets in Table 6. In the case of the local z-vector system, we used both digit-dependent priors and the component fusion technique.

		r	EER (M/F)	DCF (M/F)
1	GMM	2	4.7%/8.2%	0.195/0.336
2	JDB	2	4.3%/6.1%	0.196/0.288
3	HSB	0.25	3.5%/4.9%	0.159/0.245
4	HSB	0.5	3.4%/4.7%	**0.148/0.234**
5	HSB	1	**3.3%/4.6%**	**0.148/0.234**
6	HSB	2	**3.3%/4.6%**	0.151/0.240

Table 4: Results on the development set obtained with 512 Gaussians and global z-vectors.

	r	EER (M/F)	DCF (M/F)
JDB	2	3.9%/5.2%	0.184/0.259
HSB	0.125	4.0%/4.5%	0.178/0.232
HSB	0.25	**3.8%/4.4%**	0.171/0.220
HSB	0.5	3.9%/**4.4%**	**0.169/0.218**
HSB	1	4.0%/4.5%	0.171/0.224
HSB	2	4.0%/4.9%	0.178/0.234
HSB	4	4.3%/5.5%	0.189/0.252
HSB	8	4.6%/5.9%	0.200/0.267
HSB + fusion	0.5	**3.6%/3.9%**	**0.152/0.197**

Table 5: Results on the development set obtained with 512 Gaussians, local z-vectors and digit-dependent backends

5. Conclusion

Using the RSR Part III development and evaluation sets as a test bed, we have shown that modeling the uncertainty in the point estimates of supervector sized features used for text-dependent speaker recognition can produce substantial gains in performance. We obtained error rate reductions of up to 25% in the case of global z-vectors (Tables 1, 4 and 7); for local z-vectors, the improvements were less dramatic but they were consistent across all experiments (Tables 3 and 5). Unlike the I-Vector Backend (whose run time computational requirements are equivalent to an i-vector extraction per trial), the Hidden Supervector Backend can be configured in a way that makes reasonable computational demands (Section 3.2). Thus the Hidden Supervector Backend can claim to be a practical solution to the problem of uncertainty modeling in text dependent speaker recognition.

		EER (M/F)	DCF (M/F)
dev	local	3.7%/3.8%	0.149/0.193
dev	global	3.2%/4.5%	0.148/0.232
dev	fusion	**2.9%/3.6%**	**0.131/0.186**
eval	local	2.6%/4.5%	0.134/0.211
eval	global	2.7%/4.7%	0.140/0.236
eval	fusion	**2.3%/4.0%**	**0.122/0.192**

Table 6: Results on the development and evaluation sets obtained with local and global Hidden Supervector systems.

6. References

[1] S. Cumani, O. Plchot, and P. Laface, “On the use of i-vector posterior distributions in PLDA,” *IEEE Trans. ASLP*, vol. 22, no. 4, 2014.

[2] P. Kenny, T. Stafylakis, *et al.*, “PLDA for Speaker Verification with Utterances of Arbitrary Duration,” *ICASSP* 2013.

[3] S. Cumani and P. Laface, “Generative Pairwise Models for Speaker Recognition,” *Odyssey Workshop* 2014.

[4] T. Stafylakis, P. Kenny *et al.*, “Speaker and channel factors in text-dependent speaker recognition,” *IEEE Trans. ASLP*, vol. 24, no. 1, 2016.

[5] A. Larcher, K. A. Lee, B. Ma, and H. Li, “Text-dependent speaker verication: Classifiers, databases and RSR2015,” *Speech Communication*, vol. 60, 2014.

[6] T. Stafylakis, P. Kenny *et al.* “Text-dependent speaker recognition using PLDA with uncertainty propagation,” *Interspeech* 2013.

[7] P. Kenny, T. Stafylakis *et al.*, “An i-vector backend for speaker verification,” *Interspeech* 2015.

[8] T. Stafylakis, P. Kenny, J. Alam, and M. Kockmann, “Text-dependent speaker recognition with random digit strings,” *IEEE Trans. ASLP* 2016 (to appear).

[9] R. J. Vogt and S. Sridharan, “Explicit modeling of session variability for speaker verification,” *Computer Speech and Language*, vol. 22, no. 1, 2008.

[10] P. Kenny, T. Stafylakis *et al.*, “JFA modeling with left-to-right structure and a new backend for text-dependent speaker recognition,” *ICASSP* 2015.

[11] P. Kenny, “A small footprint i-vector extractor,” *Odyssey Workshop* 2012.

APPENDICES

A. Collecting Baum Welch Statistics with a Speech Recognition System

Given the success of phonetic DNNs in text-independent speaker recognition (where ground truth phonetic transcriptions are unavailable), it is natural to use forced alignments obtained with a speech recognition system to collect Baum-Welch statistics in text-dependent speaker recognition (where ground truth phonetic transcriptions

	EER (M/F)	DCF (M/F)
1	3.9%/4.8%	0.172/0.283
2	4.7%/5.2%	0.188/0.294
3	5.6%/6.1%	0.252/0.336
4	**3.5**%/4.1%	0.166/0.246
5	**3.5**%/**4.0**%	**0.152/0.197**

Table 7: Results on the development set obtained with the Hidden Supervector Backend when Baum-Welch statistics are collected by forced alignment.

are available). This leads to another way of implementing the Hidden Supervector Backend; results are reported in Table 7. Note that although the features here are global rather than local $\boldsymbol{z}$-vectors, using forced alignments implicitly models the left-to-right structure in the data. The results in line 1 of Table 7 were obtained with a speech recognition system trained on conversational telephone speech; the number of active senones in the digit vocabulary was found to be 276. We replicated this experiment in line 2, suppressing the covariance term in (10). Note that this leads to a substantial degradation. Training the decision tree on the RSR Part III background data turns out to be susceptible to over fitting: 180 senones (line 4) gives much better results than 528 senones (line 3). Line 5 replicates the experiment in line 3 using full rather than diagonal covariance matrices to whiten the Baum-Welch statistics. The results in line 5 of Table 7 are as good as those in the last line of Table 5 (and they have the advantage of being obtained without the component fusion technique which requires a development set in order to estimate the fusion weights).

B. I-Vector Extractors with Non-Standard Priors

The I-Vector Backend and the Hidden Supervector Backend can be treated in a unified way by taking the number of mixture components, C, to be twice the number mixture components in the UBM in the former case and $C = 2$ in the latter. We denote the zero and first order statistics associated with a mixture component c by N_c and $\boldsymbol{F}_c$. We assume that these are pre-whitened as in [11].

We postulate a hidden variable $\boldsymbol{w}$ of dimension $R \times 1$ and we interpret the Baum-Welch statistics associated with the mixture component c as summaries of a collection of noisy observations $\boldsymbol{O}$ of $\boldsymbol{T}_c\boldsymbol{w}$. The prior on $\boldsymbol{w}$ is assumed to be Gaussian with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{P}$; we denote it by $P(\boldsymbol{w})$. We evaluate the probability of $\boldsymbol{O}$ (the "evidence") by integrating out the hidden variable:

$$P(\boldsymbol{O}) = \int P(\boldsymbol{O}|\boldsymbol{w})P(\boldsymbol{w})\mathrm{d}\boldsymbol{w}.$$

Proposition 1. *The posterior distribution $Q(\boldsymbol{w})$ is Gaussian with covariance matrix $\boldsymbol{C}$ and mean $\langle \boldsymbol{w} \rangle$ given by*

$$\boldsymbol{C} = \left(\boldsymbol{P} + \sum_c N_c \boldsymbol{T}_c^* \boldsymbol{T}_c\right)^{-1} \quad (11)$$

$$\langle \boldsymbol{w} \rangle = \boldsymbol{C}\left(\boldsymbol{P}\boldsymbol{\mu} + \sum_c \boldsymbol{T}_c^* \boldsymbol{F}_c\right).$$

Proposition 2. *Letting $\langle \boldsymbol{s} \rangle = \boldsymbol{T}\langle \boldsymbol{w} \rangle$, the log evidence is given (up to irrelevant additive terms) by*

$$\sum_c \langle \boldsymbol{s}_c^* \rangle \boldsymbol{F}_c - \frac{1}{2}\sum_c N_c \langle \boldsymbol{s}_c^* \rangle \langle \boldsymbol{s}_c \rangle + \frac{1}{2}\ln|\boldsymbol{PC}| - \frac{1}{2}(\langle \boldsymbol{w} \rangle - \boldsymbol{\mu})^* \boldsymbol{P}(\langle \boldsymbol{w} \rangle - \boldsymbol{\mu}). \quad (12)$$

Proof. We use the formula for the variational lower bound on the log evidence $\ln P(\boldsymbol{O})$, namely

$$\langle \ln P(\boldsymbol{O}|\boldsymbol{w}) \rangle - D\left(Q(\boldsymbol{w}) \| P(\boldsymbol{w})\right). \quad (13)$$

Here $Q(\boldsymbol{w})$ refers to the posterior distribution of $\boldsymbol{w}$ given $\boldsymbol{O}$ and $\langle \cdot \rangle$ to the posterior expectation. The divergence can be calculated using the formula for the divergence between 2 R-dimensional Gaussians, giving

$$-\frac{R}{2} - \frac{1}{2}\ln|\boldsymbol{PC}| + \frac{1}{2}\mathrm{tr}\,(\boldsymbol{PC}) + \frac{1}{2}(\langle \boldsymbol{w} \rangle - \boldsymbol{\mu})^* \boldsymbol{P}(\langle \boldsymbol{w} \rangle - \boldsymbol{\mu}).$$

For the first term in (13), ignoring the contributions of terms which involve only the zero order and second order statistics (they are not needed to calculate evidence ratios), we can write this as

$$-\frac{1}{2}\sum_c \Big(-2\langle \boldsymbol{s}_c^* \rangle \boldsymbol{F}_c + N_c \langle \boldsymbol{s}_c^* \rangle \langle \boldsymbol{s}_c \rangle + N_c\,\mathrm{tr}\,(\boldsymbol{K}_c)\Big)$$

where, for each c, $\boldsymbol{K}_c = \mathrm{Cov}\,(\boldsymbol{s}_c, \boldsymbol{s}_c)$ so that $\mathrm{tr}\,(\boldsymbol{K}_c) = \mathrm{tr}\,(\boldsymbol{T}_c^*\boldsymbol{T}_c\boldsymbol{C})$. The formula (11) for $\boldsymbol{C}$ shows that when the contribution of these matrix traces is combined with the term $-\frac{1}{2}\,\mathrm{tr}\,(\boldsymbol{PC})$ in the expression for the divergence, the result reduces to $-\frac{1}{2}R$ and the required formula for the log evidence follows. □

Proposition 3. *The minimum divergence re-estimation formulas for the prior are*

$$\boldsymbol{\mu} = \frac{1}{S}\sum_s \langle \boldsymbol{w}(s) \rangle$$

$$\boldsymbol{P}^{-1} = \frac{1}{S}\sum_s \langle \boldsymbol{w}(s)\boldsymbol{w}^*(s) \rangle - \boldsymbol{\mu}\boldsymbol{\mu}^* \quad (14)$$

where S is the total number of sets of Baum-Welch statistics available for training and, for each set s, $\boldsymbol{w}(s)$ is the corresponding hidden variable.