Prosodic Speaker Verification using Subspace Multinomial Models with Intersession Compensation

Marcel Kockmann[1], Lukáš Burget[1], Ondřej Glembek[1], Luciana Ferrer[2] and Jan "Honza" Černocký[1]

[1]Brno University of Technology, Speech@FIT, Czech Republic
[2]Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, USA

Abstract

We propose a novel approach to modeling prosodic features. Inspired by Joint Factor Analysis model (JFA), our model is based on the same idea of introducing subspace of model parameters. However, the underlying Gaussian Mixture distribution of JFA is replaced by multinomial distribution to model sequences of discrete units rather than continuous features. In this work, we use the subspace model as a feature extractor for support vector machines (SVMs), similar to the recently proposed JFA in total variability space. We can show the capability to reduce high-dimensional count vectors to low dimension while keeping system performance stable. With additional intersession compensation, we can improve 30% relative to the baseline system and reach an equal error rate of 8.8% on the NIST 2006 SRE dataset.

Index Terms: speaker verification, prosody, JFA, multinomial model

1. Introduction

JFA is a model introduced to cope with the problem of speaker and session variability in Gaussian mixture model (GMM)-based speaker verification [1] and has established itself as the de-facto standard for high-performing speaker verification based on acoustic low-level features. Alternatives to the low-level features, which are mostly based on short-time Mel-cepstrum, are the high-level prosodic features like phone or syllable durations, pitch and energy contours, etc. Although high-level features perform worse on their own, they contain information complementary to low-level features, and fusion of systems based on these two kinds of features usually leads to improved performance [2]. Successful systems based on high-level features usually make use of a large number of prosodic features (see Section 3), which are a mixture of continuous and discrete values. Moreover, some of the features may have undefined values for some input frames (e.g., pitch is not defined for unvoiced speech). Therefore, it is difficult to model prosodic features using GMM-based models like JFA, which are suitable for modeling relatively low-dimensional continuous features. The approach adopted in [4], which also serves as our baseline, was to train a separate GMM for each feature. Occupation counts of all the Gaussian components in such GMM ensemble were estimated for all training and test utterances. The high-dimensional super-vectors of all the counts were used to represent the utterances and served as input to the SVM classifier.

There were successful attempts to apply JFA to prosodic features [2, 3]. However, only a small subset of well-defined continuous prosodic features could have been used with JFA. In [5], the approaches based on JFA and SVM are directly compared. Although JFA compares favorably to SVM on the subset of well-defined continuous features, a significant gain can be obtained with SVM trained on count super-vectors when prosodic features – that JFA cannot deal with – are added.

Recently, excellent results on acoustic features were obtained with a simplified variant of JFA [7], where the mean super-vector constructed by concatenating the means of all Gaussian components is constrained to live in subspace defined as

$$\mathbf{m} + \mathbf{T}\mathbf{w}, \tag{1}$$

where $\mathbf{m}$ is an origin of the super-vector subspace, columns of low-rank matrix $\mathbf{T}$ are bases spanning the subspace, and $\mathbf{w}$ are coordinates defining position of the mean super-vector in the subspace. The low-dimensional vector $\mathbf{w}$ is also known as i-vector. Parameters $\mathbf{m}$ and $\mathbf{T}$ can be estimated together with vectors $\mathbf{w}$ (one for each training utterance) to maximize likelihood of training data. After the subspace is learned, it can be used to extract vectors $\mathbf{w}$ for all enrollment and test utterances. In this approach, the JFA-like model serves only as the extractor of the vectors $\mathbf{w}$, which can be seen as low-dimensional fixed-size representations of utterances, and which are in turn used as inputs to a classifier (such as SVM). Note that unlike in the standard JFA, where two subspaces are used to account for speaker and inter-session variability, this simplified JFA variant uses a single subspace accounting for all the variability. Therefore, the extracted vectors $\mathbf{w}$ are not free of channel effect and inter-session compensation must be eventually considered during classification. A simple but effective channel compensation for i-vectors was proposed in [7]: it is based on feature normalization called within-class covariance normalization (WCCN) [8].

In our proposed approach, we combine the advantage of the JFA-like subspace model with the flexibility of representing prosodic features as the super-vector of occupation counts. Since the occupation counts can be seen as counts of discrete events - a component generating a frame - the process of their extraction can be seen as discretization of the original prosodic features. Therefore, as a generative model, multinomial distribution would appear as a natural choice for modeling such

This work was funded by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the Army Research Laboratory (ARL). All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI, or the U. S. Government. The work was also partly supported by European project MOBIO (FP7-214324), Grant Agency of Czech Republic project No. 102/08/0707, Czech Ministry of Education project No. MSM0021630528 and by BUT FIT grant No. FIT-10-S-2. Marcel Kockmann was supported by SVOX Deutschland GmbH.

counts.[1] In our model, the super-vector of model parameters is also constrained to live in a subspace defined by (1). However, the super-vector of Gaussian means is replaced by a super-vector of log probabilities, which are the natural parameters of our underlying multinomial distribution. A similar idea of subspace modeling of multinomial distribution was proposed for inter-session variability compensation in phonotactic language identification in [6]. A similar model is also applied for modeling GMM weights in subspace GMM, which is a recently proposed acoustic model for speech recognition [9].

2. Multinomial Subspace Model

2.1. Likelihood function

The log-likelihood of data $\mathcal{D}$ for a multinomial model with C discrete classes is determined by model parameters ϕ and sufficient statistics γ, representing the occupation counts of classes for all N utterances in $\mathcal{D}$:

$$\log p(\mathcal{D}) = \sum_{n=1}^{N}\sum_{c=1}^{C}\gamma_{nc}\log\phi_{nc}, \tag{2}$$

where γ_{nc} is the occupation count for class c and utterance n and ϕ_{nc} are probabilities of (utterance dependent) multinomial distribution, which is defined by a subspace model according to Equation 1:

$$\phi_{nc} = \frac{\exp(m_c + \mathbf{t}_c\mathbf{w}_n)}{\sum_i^C \exp(m_i + \mathbf{t}_i\mathbf{w}_n)}, \tag{3}$$

where $\mathbf{t}_c$ is the c-th row of subspace matrix $\mathbf{T}$ and $\mathbf{w}_n$ is an r-dimensional column vector (i-vector) representing speaker and channel of utterance n.

2.2. Parameter reestimation

The model parameters are obtained by maximum likelihood (ML) estimation. First, the subspace parameters $\mathbf{m}$ and $\mathbf{T}$ need to be estimated from training data. This is an iterative process, where we alternate between estimating subspace parameters $\mathbf{m}$ and $\mathbf{T}$ with fixed i-vectors, and estimating i-vectors $\mathbf{w}_n$ (one for each training utterance) with fixed subspace parameters. Even with fixed subspace parameters, there is no closed-form solution for ML update of i-vectors, and each i-vector must be updated using an nonlinear optimization technique, which is again an iterative procedure. Likewise, there is no closed-form solution for ML update of subspace parameters with fixed i-vectors. The updates we have adopted in our implementation are based on updates used for subspace GMM [9]. Vectors $\mathbf{w}_n$ are updated as

$$\mathbf{w}_n^{new} = \mathbf{w}_n^{old} + \mathbf{H}_n^{-1}\mathbf{g}_n, \tag{4}$$

where $\mathbf{g}_n$ is the gradient of the log likelihood function

$$\mathbf{g}_n = \sum_{i=1}^{C}\mathbf{t}_i^T(\gamma_{ni} - \phi_{ni}^{old}\sum_j^C\gamma_{nj}) \tag{5}$$

and $\mathbf{H}_n$ is an $r \times r$ matrix

$$\mathbf{H}_n = \sum_{i=1}^{C}\mathbf{t}_i^T\mathbf{t}_i\max(\gamma_{ni}, \phi_{ni}^{old}\sum_j^C\gamma_{nj}), \tag{6}$$

where ϕ_{ni}^{old} refers to the multinomial distribution (3) defined by the parameters from the preceding iteration. Note that the matrix $\mathbf{H}_n$ can be interpreted as an approximation to the Hessian matrix and the update formula (4) can be then seen as a Newton-Raphson update. The rows of matrix $\mathbf{T}$ are updated as

$$\mathbf{t}_c^{new} = \mathbf{t}_c^{old} + \mathbf{H}_c^{-1}\mathbf{g}_c, \tag{7}$$

where $\mathbf{g}_c$ is the gradient of the log likelihood function

$$\mathbf{g}_c = \sum_{n=1}^{N}(\gamma_{nc} - \phi_{nc}^{old}\sum_{i=1}^{C}\gamma_{ni})\mathbf{w}_n^T \tag{8}$$

and $\mathbf{H}_c$ is an $r \times r$ matrix

$$\mathbf{H}_c = \sum_{n=1}^{N}\max(\gamma_{nc}, \phi_{nc}^{old}\sum_{i=1}^{C}\gamma_{ni})\mathbf{w}_n\mathbf{w}_n^T. \tag{9}$$

The updates for both $\mathbf{w}_n$ and $\mathbf{T}$ may fail to improve likelihood by making too large an update step. In the case of such failure, we start halving the update step until an increase in likelihood is obtained. We have not provided any formula for updating vector $\mathbf{m}$. However, this can be simulated by fixing one of the coefficients in vectors $\mathbf{w}_n$ to be one and regarding the corresponding column of matrix $\mathbf{T}$ as the vector $\mathbf{m}$.

So far, we considered only subspace modeling of single multinomial distribution in our equations. However, for the prosodic features extracted by the ensemble of GMMs, the occupation counts should be modeled by a set of multinomial models, one for each GMM. We consider these to be concatenated into single super-vector of multinomial distributions, which is modeled by one subspace matrix $\mathbf{T}$. In other words, there will be only one i-vector $\mathbf{w}_n$ defining the whole set of multinomial distributions for each segment n. To achieve this, the indices c from Equation (3) must be divided into subsets, where each subset corresponds to mutually exclusive events (counts from one GMM). Then, the only difference will be in the denominator of (3), where we normalize only over the appropriate subset of indices that the current c belongs to. After the subspace parameters are estimated on training data, the model can be used to extract i-vectors $\mathbf{w}_n$ for all enrollment, test and background utterances using the same update formulae (4-6).

2.3. Model initialization

While Section 2.2 is quite general, the model initialization is described here more specifically for the used system. First, we estimate multinomial distributions for individual GMMs from the ensemble using all training utterances. This corresponds to summing all training super-vectors of occupation counts and normalizing the resulting super-vector over the ranges corresponding to individual GMMs. We will denote such super-vector of multinomial distributions as $\mathbf{sv}_{UBM}$. The vector $\mathbf{m}$ is simply initialized to a log of $\mathbf{sv}_{UBM}$. Note that we did not observe any advantage from its further retraining using the updates from the previous section. All vectors $\mathbf{w}$ are initialized with zero. To ensure a good starting point, the subspace matrix $\mathbf{T}$ is initialized to represent the most important directions in the space of model parameter super-vectors. $\mathbf{T}$ is initialized by eigenvectors of covariance matrix computed from smoothed utterance super-vectors $\mathbf{sv}_n$ centered around the vector $\mathbf{m}$. The vectors $\mathbf{sv}_n$ are computed per component as

$$sv_{nc} = \log(\alpha\frac{\gamma_{nc}}{f_{nc}} + (1-\alpha)sv_{UBM_{nc}}), \tag{10}$$

[1] More precisely, there would be a set of multinomial distributions, one for each GMM in the ensemble. For each frame, each GMM is expected to generate a feature by one of its components. This corresponds to co-occurring events that has to be modeled by separate multinomial distributions.

where f_{nc} is the number of feature frames seen for the utterance and for the GMM that the occupation count γ_{nc} corresponds to. The smoothing constant $\alpha = 0.9$ ensures that we do not take log of zero for classes that have not been occupied at all by any frames of utterance n.

3. Baseline System

Our underlying baseline system is a remake of the SRI SNERF system as described in detail in [4]. Here, we describe details of our current implementation.

3.1. Features

We use SNERFs, which are syllable-based prosodic features based on estimated F0, energy, and duration information. Characteristics like minimum, maximum, mean, and slope of pitch and energy trajectories are extracted for each detected syllable in an utterance and its nucleus, as well as duration of onset, nucleus and coda of the syllable. All values are further normalized with different techniques and form several hundred features for each syllable. The used syllable segmentation is generated from the output of a large vocabulary continuous speech recognition (LVCSR) system using human-created rules, where the phone alignments of the recognized words are used to generate correct English syllables. Detailed information on SNERFs is given in [10]. We use 142 basic features that are extracted for each syllable. Furthermore, temporal dependencies are modeled by constructing small vectors concatenating features from consecutive syllables and pauses. These so-called tokens are formed for each basic feature by concatenating up to three values (feature values and duration of pauses, more details are also given in [4]). Nine different n-gram tokens are used.

3.2. GMM-based extractor of count super-vectors

For each basic feature and token, we train a separate GMM with a small number of mixture components, instead of one large GMM for the concatenated feature vector, which would comprise a huge number of parameters. However, this way we do not model relations between the features. Each GMM is initialized with one component and these are split during expectation-maximization (EM) training until the final number g_c of mixtures is reached. g_c is determined by the number of frames that have been seen for each token, so uni-gram GMMs will have more components than, e.g., trigram GMMs. As basic features may be undefined, e.g., when no pitch is detected or when the syllable lacks onset or coda, a special GMM is needed, using an additional parameter for probability of a feature being undefined. In the first pass, all GMMs are trained to their final size using frames with defined features only, where the additional parameter is set to unit and the model falls back to a standard GMM. The GMMs are then retrained with all feature vectors, allowing the new parameter to adapt to the data. Details of the modified algorithms are given in [11]. The resulting concatenated universal background model (UBM) consists of 9×142 GMMs with different numbers of mixtures resulting in a total number of 29820 Gaussian components, where their prior probabilities define a set of multinomial models.

3.3. SVM scoring

After training the background model, we gather Gaussian component occupation counts for each training and test utterance. All counts are divided by number of frames and further rank-normalized to serve as high-dimensional input features for an SVM classifier. We train a linear kernel SVM using all background utterances as negative examples. Each speaker model SVM is then used to classify the test utterance counts, and the output is used as a decision score.

Table 1: *EER and DCF for GMM-SVM baseline system.*

Norm	Overall		Males		Females	
	EER	DCF	EER	DCF	EER	DCF
none	13.27	5.32	13.43	4.78	13.23	5.68
t-norm	**12.68**	**5.06**	**12.74**	**4.53**	**12.43**	**5.35**

4. Experiments

4.1. Data

Experiments are performed on the core condition of the NIST 2006 Speaker Recognition Evaluation (SRE), which contains English trials only. The 1-side training 1-side test condition is considered, where approximately 2.5 minutes of speech (from a 5-minute telephone conversation) are available to train each speaker and for each test utterance. This set contains 329 female and 248 male training utterances (where multiple utterances can arise from one distinct speaker), 1846 target trials, and 21841 nontarget trials. Results are presented in terms of equal error rate (EER [%]) and decision cost function×100 (DCF). The UBMs as well as the total variability subspaces, the within-class covariance and the linear discriminant analysis (LDA) matrices are trained on all-English one-conversation utterances from the NIST 2004 and 2005 SRE data sets. A set of 150 z- and t-norm utterances, respectively, per gender is taken randomly from NIST 2004 and 2005 databases, where speakers are distinct.

4.2. Baseline system

The SNERF GMM-SVM system is used as described in Section 3 and t-norm is applied to the scores. Without score normalization, we get an EER of 13.3% and t-norm drops the EER to 12.7%, as shown in Table 1. Our results also correspond with numbers for the reference system reported in [4].

4.3. Subspace model

First, we use the alternate training of $\mathbf{w}$ and $\mathbf{T}$ according to Section 2 in three iterations to enhance likelihood on the training data. Once $\mathbf{T}$ is trained, we estimate vectors $\mathbf{w}$ for all background, training, and test utterances in one iteration. Vectors $\mathbf{w}$ are used as input for an SVM with cosine kernel according to [7]. We did not see better accuracy than for the linear kernel, but we can omit the rank normalization, which was used in the baseline system with the linear kernel. The cosine kernel is used also for all following experiments. Also, we did not see any significant change in EER with t-norm applied to the scores of the subspace model, so all results reported for this model are without t-norm. Figure 1 shows the trend of EER for different numbers of factors r. While the performance is bad with small number of factors, the EER decreases and converges quite quickly and indicates that the proposed approach is indeed working. Interestingly, with 250 factors we get better accuracy than the baseline system for males (12.4% EER), but worse for females (14.7% EER) (EER of the baseline system in Table 1 is slightly better for the female than for males). Another interesting property is that we obtain reasonably good performance

Figure 1: *Dependence of EER on size of subspace.*

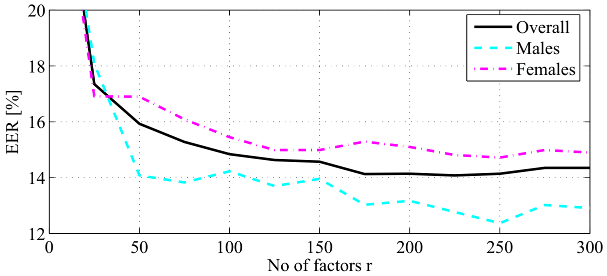

Table 2: *EER and DCF for subspace system with 250 factors and different kernel types.*

Kernel	Overall		Males		Females	
	EER	DCF	EER	DCF	EER	DCF
Cosine	14.14	5.34	12.37	4.64	14.72	5.73
WCCN	11.32	**4.61**	9.98	**3.91**	11.86	**5.01**
LDA(75)	9.91	4.90	9.44	4.28	10.33	5.26
LDA(75) + fast Sc.	**8.83**	4.86	**9.04**	4.28	**8.69**	5.25

when the subspace $\mathbf{T}$ is initialized purely with principal component analysis (PCA). In preliminary experiments, we saw a absolute loss of only 1% in EER with this highly simplified approach.

4.4. Intersession compensation

On top of our low-dimensional input vectors, we perform intersession compensation using the methods described in [7], based on our best-performing subspace system with 250 factors. First, we apply WCCN directly on our low-dimensional i-vectors. As depicted in Table 2 we get a relative improvement of 20% in EER and 14% in DCF, respectively, due to the WCCN. Alternatively, we use LDA to both, diagonalize the across-class-covariance matrix and make the within-class-covariance matrix the identity matrix. LDA can be used to further reduce the feature dimension by dropping the nuisance directions that correspond to channel. As shown in Table 2, dropping nuisance dimensions seems to help a lot and we get the best error rate of 9.9% using LDA(75) reduction to 75 dimensions. Finally, we use a simplification of the speaker enrollment and testing procedure, also successfully applied in [7], where the value of the SVM kernel function evaluated for enrollment and test utterance is directly taken as the score. This greatly reduces the computational complexity, as no SVM training is needed and only a dot-product is computed during testing. As shown in the last line of Table 2, we get a further improvement of 11% relative to an excellent EER of 8.8% with this fast scoring technique. Also, the observed difference between the male and female subsystems seems to vanish. This may indicate that SVM parameters could have been optimized individually for the two classifiers. Note that we have to apply zt-norm to these scores to obtain good performance, while we did not see any positive effect of zt-norm for the SVM-based approaches. Another interesting phenomenon we observed in our experiments is that while EER was consistently better for a smaller number of factors (50-100 after LDA), the DCF was generally better for a larger number of factors. As depicted in bold numbers in Table 2, the best overall DCF×100 of 4.61 was observed for a system with 250 factors, no LDA and WCCN only.

5. Conclusions and Lookout

We have proposed a novel subspace modeling approach for multinomial models in speaker verification, where the basic assumption of JFA based on GMMs is successfully transferred to multinomial models. We have shown the capability of the approach to reduce very high-dimensional input features to several hundred dimensions, while preserving their discriminative power. Furthermore, intersession compensation in low-dimensional subspace reduces error rate significantly and our approach outperforms the baseline system by 30% relative. On one hand, the gain through intersession compensation is less than reported in [7] for a GMM system with low-level short time features, but on the other hand, [5] states that intersession variability compensation does not lead to any improvements on the prosodic high-dimensional feature vectors (as used in Section 4.2). We can conclude that the reduction of the high-dimensional vectors to several hundred dimensions allows the channel compensation to work. Generally, the computational efforts of the proposed approach mainly lie in the background model training procedure. The size of the SVM classifiers that must be evaluated during the training and testing phase is reduced by a factor of more than 100. Furthermore, SVM training can be omitted and replaced with the fast scoring technique.

Following this approach, we are already working on applying distinct speaker and channel subspaces to the multinomial model, as is the basic assumption in the standard JFA model, followed by standard log-likelihood-ratio scoring. In addition, we are considering the introduction of a prior on the distribution of the i-vectors $\mathbf{w}$ as found in the standard JFA model.

6. References

[1] Kenny, P., et al., “A Study of Inter-Speaker Variability in Speaker Verification”, IEEE Trans. Audio, July 2008, Vol. 16, p. 980-988.

[2] Dehak, N., et al., “Modeling Prosodic Features with Joint Factor Analysis for Speaker Verification”, IEEE ASLP, Sept. 2007, pp. 2095–2103.

[3] Kockmann, M., et al., “Investigations into Prosodic Syllable Contour Features for Speaker Recognition”, in Proc. ICASSP, Dallas, Mar. 2010.

[4] Ferrer, L., et al., “Parameterization of Prosodic Feature Distributions for SVM Modeling in Speaker Recognition”, in Proc. ICASSP, Honolulu, April 2007.

[5] Ferrer, L., et al., “A Comparison of Approaches for Modeling Prosodic Features in Speaker Recognition”, in Proc. ICASSP, Dallas, Mar. 2010.

[6] Glembek, O., et al., “ Advances in Phonotactic Language Recognition”, in Proc. Interspeech, Brisbane, Sept. 2008.

[7] Dehak, N., et al., ”Front-End Factor Analysis for Speaker Verification”, submitted to IEEE ASLP, November 2009.

[8] Hatch, A. O., et al., “Within-class covariance normalization for SVM-based speaker recognition,” in Proc. ICSLP, Pittsburgh, Sept. 2006.

[9] Povey, D., et al., “Subspace Gaussian Mixture Models for Speech Recognition”, in Proc. ICASSP, Dallas, Mar. 2010.

[10] Shriberg, E., et al., “Modeling Prosodic Feature Sequences for Speaker Recognition”, Speech Comm. 46, July 2005, pp. 455–472.

[11] Kajarekar, S., et al., “Modeling NERFs for Speaker Recognition”, in Proc. Odyssey 04, Toledo, Spain.