Using Deep Belief Networks for Vector-Based Speaker Recognition[†]

W. M. Campbell

MIT Lincoln Laboratory, Lexington, MA, USA

wcampbell@ll.mit.edu

Abstract

Deep belief networks (DBNs) have become a successful approach for acoustic modeling in speech recognition. DBNs exhibit strong approximation properties, improved performance, and are parameter efficient. In this work, we propose methods for applying DBNs to speaker recognition. In contrast to prior work, our approach to DBNs for speaker recognition starts at the acoustic modeling layer. We use sparse-output DBNs trained with both unsupervised and supervised methods to generate statistics for use in standard vector-based speaker recognition methods. We show that a DBN can replace a GMM UBM in this processing. Methods, qualitative analysis, and results are given on a NIST SRE 2012 task. Overall, our results show that DBNs show competitive performance to modern approaches in an initial implementation of our framework.

Index Terms: speaker recognition, deep belief networks

1. Introduction

Deep Belief Networks (DBNs) have become a popular research area in machine learning [1] and in acoustic modeling for large-vocabulary automatic speech recognition (ASR) [2, 3]. This research significantly expands upon earlier work on multilayer perceptrons (MLPs) and their applications to speech processing; see, for example [4]. Earlier work in MLPs focused on "shallow" architectures—either one or two layers—that were discriminately trained using labeled classes. The breakthrough for DBNs has been a combination of the development of unsupervised methods (pretraining), GPU-based acceleration, and deeper architectures. Although one-layer MLPs are known to be universal approximators [5], they require, potentially, a large number of hidden units. DBNs, in contrast, use a powerful strategy of unsupervised training and multiple layers that provides parameter-efficient and accurate acoustic modeling [3].

Speaker recognition research is fundamentally different from approaches in automatic speeech recogntion since the focus in speaker recognition has been on vector-based approaches such as GMM supervectors [6] and the more recent i-vectors [7]. In these approaches, a GMM universal background model (GMM UBM) is used to derive a vector representation of an utterance which is then used for speaker modeling. Much of the focus of this work has been on strategies for vector classification—simple inner product methods [7], probabilistic linear discriminant analysis (PLDA) [8], support vector machines [6], and advanced Bayesian methods [9]. Also, methods such as WCCN [10], NAP [11], and PLDA [8] for compensating and modeling of speaker session and channel variation have been key topics.

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

Initial approaches to using DBNs and restricted Boltzmann machines (RBMs) for speaker recognition have appeared in [12] and [13]. The first paper focuses on modeling of i-vectors using RBMs. The second approach models the statistics of the output of a DBN by i-vectors. Both of these approaches show a growing interest and moderate success in applying DBNs to speaker recognition.

Our new approach to using DBNs for speaker recognition is to replace a GMM UBM with a DBN in the early acoustic modeling stages. We use the output of a DBN, which is a posterior probability, as a substitute for the GMM UBM mixture component posterior probability which is commonly used in GSV and i-vector systems. Similar to ASR approaches, we use pretraining methods to model at the frame level and explore multi-layer architectures. Additionally, we explore the role of sparsity and selectivity in the output of the DBN and its effect on performance.

The outline of the paper is as follows. In Section 2, we describe DBNs and standardize notation for methods used in the rest of the paper. Section 3 reviews standard vector-based speaker recognition methods. Next, in Section 4, we detail our approach for using DBNs in speaker recognition. Section 5 describes experiments using DBNs for modeling, describes some intuition behind the process, and details experiments on the NIST SRE 2012 evaluation using our methodology.

2. Deep Belief Networks

DBNs have been successfully used in speech recognition for modeling the posterior probability of state given a feature vector [3], $p(q_t|\mathbf{x}_t)$. Feature vectors are typically standard frame-based acoustic representations (e.g., MFCCs) that are usually stacked across multiple frames. A basic training strategy to estimate this posterior involves multiple phases. First, pretraining of the DBN is accomplished by successively training restricted Boltzmann machines and stacking them. Second, optimization with backpropagation—typically, referred to as fine-tuning—using labels from an ASR is used to discriminately train the DBN. Third, the resulting models can be used for realignment and retraining—e.g., embedded Viterbi. Many variants of this procedure have been proposed. Note that we use the term DBN in this paper to encompass all of the different variants—pretrained, pretrained and then fine-tuned, and trained from a random start with standard backpropagation methods.

Since in text-independent speaker recognition time information is ignored, only the first two steps are appropriate for our purposes. In this section, we focus on pretraining for DBNs. The basic process for pretraining a DBN is based upon stacking RBMs. RBMs are an undirected graphical model with visible and hidden units with only visible hidden connections. For initial discusions, we assume both the hidden and visible units are Bernoulli distributed. The parameters of the RBM are given by, $\mathbf{W}$ (visible/hidden connection weights), $\mathbf{b}$ (visible-unit bias),

and $\mathbf{c}$ (hidden-unit bias). We use column vectors for $\mathbf{b}$ and $\mathbf{c}$ and define $\mathbf{W}$ to have dimension N_h (number of hidden units) by N_v (num of visible units).

Optimization of the parameters is performed by first defining the energy function,

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{b}^t\mathbf{v} - \mathbf{c}^t\mathbf{h} - \mathbf{h}^t\mathbf{W}\mathbf{v}. \quad (1)$$

The corresponding probability of a configuration is given by,

$$P(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{Z} \quad (2)$$

where Z is a normalizing factor obtained by summing the numerator of (2) over all possible states of $\mathbf{h}$ and $\mathbf{v}$.

As noted by many authors, optimization of the model parameters $(\mathbf{W}, \mathbf{b}, \mathbf{c})$ using maximum likelihood methods and the probability (2) is difficult because of the complexity of computing the normalizing factor Z. A 1-step contrastive divergence method is used instead. Conditional probabilities for (2) are given by,

$$P(\mathbf{h} = \mathbf{1}|\mathbf{v}) = \sigma(\mathbf{W}\mathbf{v} + \mathbf{c}) \quad (3)$$

and

$$P(\mathbf{v} = \mathbf{1}|\mathbf{h}) = \sigma(\mathbf{W}^t\mathbf{h} + \mathbf{b}). \quad (4)$$

The approximation of the gradient using the contrastive divergence method is given by

$$-\frac{\partial L}{\partial \mathbf{W}} \approx \langle \mathbf{h}_0 {\mathbf{v}_0}^t \rangle - \langle \mathbf{h}_1 \mathbf{v}_1^t \rangle \quad (5)$$

where L is the likelihood of the training data using the probability (2) and the angle brackets, $\langle\rangle$, indicate an average over a training set. The quantity $\mathbf{v}_0$ is a training data exemplar and $\mathbf{h}_0$ is calculated from (3). The quantities $\mathbf{v}_1$ and $\mathbf{h}_1$ are found using one step of Gibbs sampling [14]. We note that, strictly speaking, the quantity on the right hand side of (5) is not a gradient—a more rigorous discussion can be found in [15].

The contrastive divergence approximation method is used in a standard gradient descent with mini-batches. For optimization, we use the update

$$\Delta\mathbf{W}_{n+1} = m\Delta\mathbf{W}_n - \alpha\frac{\partial L}{\partial \mathbf{W}} \quad (6)$$

where m is a momentum term and α is the learning rate.

We found it useful (for reasons explained in the next section) to encourage sparsity in the output of the RBM. Multiple methods are available for incorporating a sparsity penalty in the RBM optimization process [14, 16, 17]. One natural approach is to use the entropy of the hidden units for a training set; this criterion would tend to encourage sparsity since the entropy function is minimized by sparse activations. Unfortunately, the derivative of this penalty term is difficult, so it is instead common to use cross-entropy with a target probability p_t. The penalty term then becomes,

$$\lambda \sum_{i\in\text{train}} \sum_{j\in\text{hidden}} p_t \log_2 h_{0,j}^{(i)} + (1 - p_t)\log_2(1 - h_{0,j}^{(i)}) \quad (7)$$

where λ is a scaling on the penalty term and $h_{0,j}^{(i)}$ is the jth entry of $\mathbf{h}_0^{(i)}$ (as in (5)) calculated from the ith element of the training set.

The sparse penalty term in (7) can be incorporated in the contrastive divergence optimization (5) by modifying the first term in the equation to be,

$$\langle \mathbf{q}_0 {\mathbf{v}_0}^t \rangle - \langle \mathbf{h}_1 \mathbf{v}_1^t \rangle \quad (8)$$

where

$$\mathbf{q}_0 = \phi p_t \mathbf{1} + (1 - \phi)\mathbf{h}_0 \quad (9)$$

and $0 \leq \phi \leq 1$, see [14, 16] for more details.

The posterior probabilities in (3) and (4) assume that both $\mathbf{x}$ and $\mathbf{h}$ have a Bernoulli distribution. For cepstral inputs to the DBN, a better assumption is that the visible layer has a Gaussian distribution. In this case, the posterior (4) becomes a normal distribution, and $\mathbf{v}_1$ is replaced by $\mathbf{W}^t\mathbf{h}_0 + \mathbf{b}$ in both (5) and (8).

3. Vector-Based Speaker Recognition

We used the term *vector-based speaker recognition* to encompass methods which take speech utterances as input and then find a vector-representation based upon the statistics of a GMM universal background model (GMM UBM). Standard methods in this area are the early GMM supervector techniques (GSV) [6] and the more recent i-vector approaches [7, 18].

Both the GSV and i-vector approaches rely upon a two-stage process. For the first stage, sufficient statistics are calculated from a sequence of feature vectors, $\{\mathbf{x}_t\}$, $t = 1, \ldots, N_t$,

$$\mathbf{s}_0 = \sum_t \left[p(1|\mathbf{x}_t) \cdots p(N_m|\mathbf{x}_t)\right]^t \quad (10)$$

$$\mathbf{s}_1 = \sum_t \left[p(1|\mathbf{x}_t)\mathbf{x}_t \cdots p(N_m|\mathbf{x}_t)\mathbf{x}_t\right]^t \quad (11)$$

$$\mathbf{s}_2 = \sum_t \left[p(1|\mathbf{x}_t)\mathbf{x}_t * \mathbf{x}_t \cdots p(N_m|\mathbf{x}_t)\mathbf{x}_t * \mathbf{x}_t\right]^t \quad (12)$$

where $p(i|\mathbf{x}_t)$ is the posterior probability of a mixture component, N_m is the number of mixture components, and '$*$' indicates element-wise multiplication.

For the GSV approach, a supervector is then calculated by using MAP adaptation to find a mean supervector. This mean vector is then transformed (e.g., by NAP and KL-divergence motivated weighting [6]) to obtain a supervector that can be used in standard classification approaches—inner-product based, SVMs, etc. For the i-vector approach, dimension reduction to a set of factors is used [7]. Using a total variability space representation $\mathbf{T}\mathbf{y} + \mathbf{m}_0$ and a Gaussian assumption, an estimate $\hat{\mathbf{y}}$ of $\mathbf{y}$ is found. The resulting factors are whitened, length-normalized, and used as inputs to inner-product, PLDA, or other backend-classifiers [7, 18].

4. DBNs for Speaker Recognition

We combine DBNs with vector-based methods by first geometrically interpreting the standard GMM-based approaches in Section 3. In these approaches, the set of posterior probabilities of the mixture component $\{p(i|\mathbf{x})\}$ act as a partition of unity of the input feature space (in the topological sense, see, e.g., [19]). This comment is just another way of saying that

$$\sum_{i=1}^{N_m} p(i|\mathbf{x}) = 1. \quad (13)$$

Another important point is that the posterior probabilities $p(i|\mathbf{x})$ are "localized" in the sense that, as a function of $\mathbf{x}$, they are large near the ith mixture mean and small near the jth mixture mean for $j \neq i$. This qualitative analysis applies in the main mass of the feature space. We note that outside of this region, in "overflow" areas of the space (in vector quantization terms), the tails of the exponential distributions dominate.

The interpretation, then, of the statistics in (10) as they are used in speaker recognition is that they are used to form local

estimates of mean shift from a global mean supervector calculated across the data. These local shifts describe how a speaker's acoustic output deviates from a reference across multiple acoustic regions in feature space. Both the partition-of-unity property and the locality of the partitioning process provide a vector description of speaker characteristics.

The straightforward method of extending this approach to DBNs is to replace the GMM UBM posterior in (10) with a DBN hidden unit generated posterior (3). Since methods for i-vectors and GSVs only rely upon these quantities, we can easily revise standard methodology to incorporate this approach. We note that in practice the output of a DBN may not be appropriately scaled, so dividing by $c_t = \mathbf{1}^t p(\mathbf{h} = \mathbf{1}|\mathbf{x}_t)$ will be necessary. Additionally, if multi-frame inputs to the DBN are used to estimate posterior in (3), then the center frame can be used in the sufficient statistics calculation.

Another important point is that the DBN output is not automatically assured to be localized. This property can be seen by considering a one layer network. If we look at the ith output in (3), then this corresponds to the ith row of $\mathbf{W}$, $\sigma(\mathbf{w}_i\mathbf{x} + c_i)$. This description shows us that we are partitioning the data into two half-spaces with a hyperplane—a non-local representation. Two factors can help increase sparsity in the output. Because of the high-dimension of the DBN processing and the sparsity of data in a high-dimensional space, we can bias DBN training as in Section 2 to create sparse outputs. Additionally, combining multiple layers of RBMs into a DBN can successively partition the space with greater locality to create a sparse output.

In addition to using sparsity in pretraining, we also explored generating sparse outputs by using labeled data. The most straightforward method borrows methodology from the ASR community. We use the most likely mixture component from a GMM UBM per frame as a sequence of labels for training the DBN. This method serves as an alternative approach for obtaining sparse outputs from a DBN.

5. Experiments

5.1. NIST SRE 2012

Our experiments are based upon the NIST 2012 Speaker Recognition Evaluation (SRE). In NIST SRE 2012, the task focused on speaker detection using multi-utterance enrollment and a variety of test conditions. Participants were given access to most of the enrollment data prior to the release of test data for development. For the purposes of this paper, we focus upon the core training condition which included both microphone and telephone data from approximately 1900 target speakers.

Multiple test conditions for the NIST SRE were available. Because of statistical significance, for our experiments we chose extended trials for the common evaluation condition. The common evaluation conditions included five different tests. To narrow the scope of our experiments, we looked at conditions 1, microphone interview speech in test, and condition 2, phone call speech in test. Both of these conditions tested performance without additive noise and are similar to prior NIST SRE evaluations. We evaluated the performance of systems using the NIST criterion, C_{primary}. The NIST criterion is obtained by averaging the performance at two operating points using a normalized cost function, C_{norm}. This cost function is defined as

$$C_{\text{norm}}(\theta) = P_{\text{miss}}(\theta) + \frac{1 - P_t}{2P_t}\left[P_{\text{fa}}(\theta|\text{known}) + P_{\text{fa}}(\theta|\text{unknown})\right]. \quad (14)$$

where θ is a threshold, P_t is the given probability of a target, P_{miss} is the miss rate, and P_{fa} is the false alarm rate from known and unknown impostors in false trials. Note that we have assumed the evaluation weighting of 0.5 for known and unknown false alarms. If we let θ_A be the threshold for $P_t = 0.01$ and θ_B be the threshold for $P_t = 0.001$, then evaluation criterion is

$$C_{\text{primary}} = 0.5\left[C_{\text{norm}}(\theta_A) + C_{\text{norm}}(\theta_B)\right]. \quad (15)$$

5.2. Baseline Systems

Baseline systems for the NIST SRE experiments were derived from participation in the NIST evaluation. Both an SVM GSV system and an i-vector system were used for experiments. Our approach to system construction was to create systems with good performance that could be adapted to our proposed DBN approach. Creating more complex models and multiple feature fusions is beyond the scope of this work (and can be a challenging task, see [20]).

In all cases, the input audio waveform was pre-processed to normalize the sampling rate to 8 kHz. Also, both noise reduction and tone-removal were performed as in [21]. Features were then generated using MFCCs plus $c0$ and associated delta-features for a total of 40 features. Feature warping was applied to the features. SAD for telephony speech was based on a combination of energy features and a GMM SAD system. SAD for microphone data was based on a two-channel strategy, see [21].

To standardize the approach, we used a GMM UBM with 512 mixture components for both the SVM GSV and i-vector systems. The expansion dimension for the SVM was as a result 20480 dimensions. For SVM training, we used a one-versus-rest approach where only target speakers were used in the training. This approach is similar to methods in language recognition and is similar to other approaches using "anti-models" in NIST SRE 2012 [22, 23, 20].

For the i-vector system, a T matrix with rank 600 was used. The resulting factors were whitened, length normalized and then WCCN was applied. Training a model was accomplished by averaging all i-vectors. Note that the reduced number of mixtures (down from a typical 2048) impacted performance. After raw scores were found for both systems, both Z- and T-norm were applied. T-norm models were based upon other target speakers. Z-norm was based on a set of utterances from NIST SRE 2005.

5.3. Training DBNs

We trained DBNs with a GPU system using contrastive-divergence and the algorithms in Section 2. For the DBN configuration, we used an input of one frame of data—a 40-dimensional input. For each layer in the DBN, we consistently used a dimension of 512 for the output. Our set of available training data set was drawn from the entire NIST SRE 2012 target data—approximately 1,300 hours of data. Using this entire data set was intractable, so we reduced data in two ways. For every epoch of processing, we only considered a fraction of the data set (typically 10%). Also, we structured the mini-batches by downsampling the feature vector set; we usually kept only every 10th vector. This reduction had the side-effect that the training data was less correlated.

For our experiments, we constructed pretrained DBNs with both 1- and 2-layers. We found that the contrastive divergence procedure was somewhat sensitive and proper care needed to be taken to obtain reasonable outputs. For our experiments, we used a learning rate of $\alpha = 0.005$, a momentum of 0.9, and approximately 30 epochs of training. Progress during training

Table 1: Performance of DBN system for various configurations on a NIST SRE evaluation task.

System	Num Layers	Pretrain (Y/N)	Sparse (Y/N)	Supervised (Y/N)	C_{primary} Mic	C_{primary} Tel
SVM GSV	-	-	-	-	0.333	0.422
i-vector	-	-	-	-	0.462	0.528
SVM DBN	1	Y	N	N	0.582	0.717
SVM DBN	1	Y	Y	N	0.483	0.638
DBN i-vector	1	Y	Y	N	0.549	0.664
SVM DBN	2	Y	N	N	0.581	0.785
SVM DBN	2	Y	Y	N	0.560	0.697
SVM DBN	2	N	-	Y	0.343	0.439
SVM DBN	3	N	-	Y	0.360	0.459
DBN i-vector	2	N	-	Y	0.451	0.523
DBN i-vector	3	N	-	Y	0.467	0.546

was monitored using standard criteria [14]. We found it useful to examine image plots of output units over time (posteriorgrams) to ensure good training. Signs of overtraining, local minima, and over-active outputs appeared as lines in the posteriorgram. We also applied the sparsity penalty as discussed in Section 2, equation (9). For this term, we found that a ϕ of 0.05 in the first stage, $\phi = 0.01$ in later stages, and a target probability $p_t = 0.002$ yielded sparser outputs in all cases. Example outputs for a 1-layer pretrained DBN are shown in Figure 1.

In addition to using pretrained DBNs, we also considered DBNs with the same architecture trained using standard backpropagation techniques. We trained systems using a scaled conjugate gradient method and a cross-entropy criterion. Labels per frame for the data were derived by taking the mixture component of the GMM UBM with the largest posterior probability. For training, a smaller subset of the data was used consisting of approximately 0.5 million randomly chosen feature vectors. Training for approximately 100 epochs was performed. As with the sparsely trained DBN, output activations were sparse when the system was trained correctly. This property successfully mimics the behavior of a GMM UBM as discussed in Section 4.

5.4. DBN Speaker Recognition Results

The baseline system performance is shown in Table 1. The i-vector performance is somewhat limited by the smaller GMM UBM (512 mixtures), but our goal was to use a uniform set of features and background models. The SVM system performs well with multi-utterance enrollment. We used the DBNs described in the last section as posterior probability generators for "sufficient" statistics as in Section 4. The statistics were used to build both SVM GSV systems and i-vector systems. Because of the tracking performance between the systems, Table 1 only documents the itermediate results for the SVM DBN system. Best performing results are shown for both the SVM and i-vector systems.

From the table, we see that the initial 1-layer pretrained DBN produced results that were reasonable but not competitive with GMM UBM systems. The next 1-layer system using sparse outputs shows a substantial drop in error rate; sparsity and the corresponding localization of statistics appears to be a useful property for speaker discrimination. Additional pretrained layers did not have any effect.

From the table, we see that supervised training using GMM UBM labels is the best approach in our current setup. As we add more layers, the performance is similar. Qualitatively, when we examined the image posteriorgram output, the DBN output had sparse activations when discriminative training was used. Note, we also tried to combine pretraining with supervised training, but we found that pretraining was not helpful.

Our experiments show that there is a mix of issues arising when using DBNs for speaker recognition. Encouraging sparsity in the output ("stars" in the posteriorgram) appears to be a desirable property. The best strategy for doing this is not obvious. In pretraining, we can encourage sparsity, but it's not clear how the optimization can be directed to produce the best recognition results. In the supervised case, sparse output using labels perform well, but the best selection of labels is not obvious. Overall, there is significant future work to explore supervised and unsupervised approaches, feature sets, and improved training methods to optimize performance.

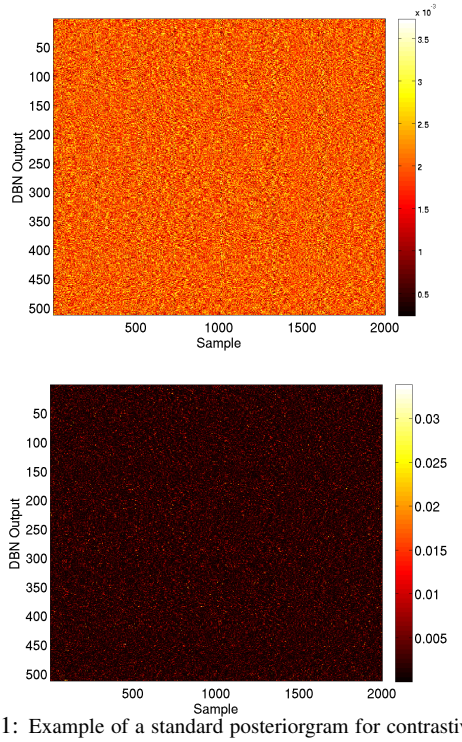

Figure 1: Example of a standard posteriorgram for contrastive divergence (top) and contrastive divergence with a sparse penalty (bottom) with a 1-layer DBN.

6. Conclusions

We have demonstrated a new method for incorporating DBNs in standard systems for speaker recognition. Our approach used a DBN in the acoustic modeling stage for generating summary statistics. Once these statistics were generated, we used vector-based speaker recognition systems for backend modeling. We demonstrated that both pretraining and supervised approaches could be used for DBNs. Evaluating on a NIST SRE 2012 task showed the viability of the new methods, but also illustrated challenges in best optimizing DBNs for this new process.

7. References

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[2] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A.-r. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 30–35.

[3] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[4] N. Morgan and H. A. Bourlard, *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[6] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proceedings of ICASSP*, 2006, pp. I–97–I–100.

[7] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, 2010.

[8] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[9] B. J. Borgstrom and A. McCree, "Supervector Bayesian speaker comparison," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7693–7697.

[10] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition." in *Interspeech*, 2006.

[11] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *Proceedings of ICASSP*, 2005.

[12] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "Preliminary investigation of Boltzmann machine classifiers for speaker recognition," in *Proceedings Odyssey Speaker and Language Recognition Workshop*, 2012.

[13] V. Vasilakakis, S. Cumani, and P. Laface, "Speaker recognition by means of deep belief networks," in *Biometric Technologies in Forensic Science Conference*, 2013.

[14] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," Machine Learning Group, University of Toronto, Tech. Rep. 2010-003, 2010.

[15] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proceedings of the tenth international workshop on artificial intelligence and statistics*. Society for Artificial Intelligence and Statistics NP, 2005, pp. 33–40.

[16] H. Goh, N. Thome, and M. Cord, "Biasing restricted Boltzmann machines to manipulate latent selectivity and sparsity," in *NIPS workshop on deep learning and unsupervised feature learning*, 2010.

[17] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area v2." in *NIPS*, vol. 7, 2007, pp. 873–880.

[18] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *Interspeech*, 2011, pp. 249–252.

[19] W. M. Boothby, *An introduction to differentiable manifolds and Riemannian geometry*. Academic press, 1986, vol. 120.

[20] R. Saeidi, K. Lee, T. Kinnunen, T. Hasan, B. Fauve, P. Bousquet, E. Khoury, P. S. Martinez, J. Kua, and C. You, "I4U submission to NIST SRE 2012: A large-scale collaborative effort for noise-robust speaker verification," in *Proc. Interspeech*, 2013.

[21] W. M. Campbell, D. Sturim, B. J. Borgstrom, R. Dunn, A. McCree, T. F. Quatieri, and D. A. Reynolds, "Exploring the impact of advanced front-end processing on nist speaker recognition microphone tasks," in *Proc. Speaker Odyssey Workshop*, 2012.

[22] H. Sun, K. A. Lee, and B. Ma, "Anti-model KL-SVM-NAP system for NIST SRE 2012 evaluation," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7688–7692.

[23] T. Hasan, S. O. Sadjadi, G. Liu, N. Shokouhi, H. Boril, and J. H. Hansen, "CRSS systems for 2012 NIST speaker recognition evaluation," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6783–6787.