Adaptation of Acoustic Models in Joint Speaker and Noise Space Using Bilinear Models

Yongwon JEONG†, Member and Hyung Soon KIM†, Nonmember

SUMMARY We present the adaptation of the acoustic models of hidden Markov models (HMMs) to the target speaker and noise environment using bilinear models. Acoustic models trained from various speakers and noise conditions are decomposed to build the bases that capture the interaction between the two factors. The model for the target speaker and noise is represented as a product of bases and two weight vectors. In experiments using the AURORA4 corpus, the bilinear model outperforms the linear model.

key words: bilinear model, eigenvoice speaker adaptation, environment adaptation, speaker adaptation, speech recognition

1. Introduction

In automatic speech recognition (ASR) systems using hidden Markov models (HMMs) [1], performance degradation is attributed to the difference between the training and test conditions. Speaker variability and environmental noise are two dominant factors that affect the performance of an ASR system. To compensate for the mismatches, adaptation techniques are used to transform generic models such as speaker-independent (SI) HMMs to the target speaker/noise in HMM-based speech recognition. Eigenvoice adaptation [2], a linear model-based adaptation technique, is based on the principal component analysis (PCA) [3] of acoustic models, and shows good performance for rapid speaker adaptation because of its small number of adaptation parameters. Thus, the bilinear model [4], the extension of the linear model to two factors, is a good candidate for the adaptation of acoustic models to both speaker and noise. In the bilinear model terminology, two factors to be decoupled are called style and content; they are mathematically interchangeable, but they usually have their own contextual meanings.

Bilinear models have been successfully used to build models with two factors. Tenenbaum and Freeman [4] propose bilinear models to model facial images of different persons (style) in different poses (content), and to model different characters (content) in different fonts (style). Mpiperis et al. [5] describe the building of a bilinear model using an ensemble of 3-D point sets that represent faces from different persons and different expressions to recognize a face and facial expression.

Bilinear models have also been investigated in the speech processing area. Song et al. [6] apply the bilinear model to speaker adaptation with two factors defined as speaker (style) and tied state (content), although only the style factor is estimated during adaptation. In Popa et al. [7], the authors apply the bilinear model to voice conversion and report good results compared with the Gaussian mixture model (GMM)-based method using a small amount of training data.

The adaptation of acoustic models to the new speaker and environment has been investigated by researchers. Wang and Gales [8] propose adapting acoustic models to the target speaker by maximum likelihood linear regression (MLLR) adaptation [9] and then to the target noise by model-based vector Taylor series (VTS) [10]. The transforms for the target speaker and noise are jointly estimated. In Seltzer and Acero [11], the authors propose using a series of constrained MLLR (CMLLR) transforms [12] to compensate for speaker and environment mismatch. The approach applies the transform for a speaker in one environment to the same speaker in a different environment. In Wang and Gales [13], factor transforms are obtained separately to compensate for multiple acoustic factors such as speaker and environment. The authors propose a method using factorized cluster adaptive training (fCAT) [14] to make the factor transforms independent and perform factorized adaptation. Saz and Hain [15] propose the factorization of speaker and background effects (e.g., background music) using CMLLR. In the approach, speaker adaptation is performed to factorize speaker and background effects, and transforms for background conditions are used asynchronously during decoding process by using the optimal transform of each frame. In Jeong [16], bases that are common across training speakers and noise conditions are constructed via the tensor decomposition of acoustic models trained from many speakers in different noise environments, and a weight vector for both speaker and noise is estimated from adaptation data.

In this letter, we present the basis-based adaptation of acoustic models to the target speaker and noise environment using the bilinear model. We propose to use the bilinear model to decouple speaker and noise variations that are present in training models and construct bases that are common across the two factors. The bases are used for the adaptation of acoustic models to the target speaker and noise by representing the updated model as a product of the bases and two weight vectors. Thus, the proposed approach is a two-
factor extension of eigenvoice adaptation, which is a repre-
sentative basis-based adaptation technique. Our method
factorizes the two factors during model building in common
with some of the techniques mentioned above. However,
such factorization has not been performed in a basis-based
framework.

This letter is organized as follows. Section 2.1 presents
the decomposition of training models into the bilinear
model, and Sect. 2.2 describes the adaptation of acoustic
models to the target speaker and noise using the bilinear
model. Section 2.3 explains the eigenvoice-based linear
model approach, with which the proposed method is com-
pared. Section 3 presents the experiments, and Sect. 4 con-
cludes the letter.

2. Methods

2.1 Construction of Bilinear Models

In this letter, the adaptation of HMMs to the target speaker
and noise is performed by updating the mean param-
eters of the output distribution among continuous density
HMM (CDHMM) parameters. Let \( \mu_r(s, n) \in \mathbb{R}^{D \times 1} \) be
the mean vector for the \( r \)th Gaussian mixture component
(\( r = 1, \ldots, R \)) from the training speaker \( s (s = 1, \ldots, S) \)
and noise \( n (n = 1, \ldots, N) \). Then, the HMM mean param-
eters of all the mixture components from the speaker \( s \) and
noise \( n \) are concatenated into an \( RD \times 1 \) vector:

\[
\mu(s, n) = \begin{pmatrix}
\mu_1(s, n) \\
\vdots \\
\mu_r(s, n) \\
\vdots \\
\mu_R(s, n)
\end{pmatrix}.
\]

Before decomposition, all HMM mean vectors are normal-
ized by subtracting the HMM mean vector of a clean speech
SI HMM:

\[
y(s, n) = \mu(s, n) - \mu_{SI}.
\]

We define the training matrix, which is an \((RD)S \times N\) matrix
containing HMM mean vectors for all training speakers and
noise conditions:

\[
Y = \begin{pmatrix}
y(1, 1) & \cdots & y(1, N) \\
\vdots & \ddots & \vdots \\
y(S, 1) & \cdots & y(S, N)
\end{pmatrix}
\]

and its vector transpose, which is an \((RD)N \times S\) matrix:

\[
Y^{VT} = \begin{pmatrix}
y(1, 1) & \cdots & y(1, N) \\
\vdots & \ddots & \vdots \\
y(S, 1) & \cdots & y(S, N)
\end{pmatrix}.
\]

The training matrix can be decomposed into the bilinear
model as follows:

\[
Y \approx (W^{VT}A)^{VT}B
\]

or \( Y^{VT} \approx (WB)^{VT}A \).

This is the symmetric bilinear model [4] and the model para-
eters are given by

\[
A = [a_1, \ldots, a_S], \quad B = [b_1, \ldots, b_N]
\]

with \( a_i \in \mathbb{R}^{I \times 1} (I \leq S - 1) \) and \( b_n \in \mathbb{R}^{J \times 1} (J \leq N - 1) \)
corresponding to the weight vectors for the speaker \( s \) and
noise \( n \), respectively, and the basis matrix \( W \) contains
the common bases that are independent of the speaker and noise
variations:

\[
W^{RD \times J} = \begin{pmatrix}
w_{11} & \cdots & w_{1J} \\
\vdots & \ddots & \vdots \\
w_{1J} & \cdots & w_{1J}
\end{pmatrix}.
\]

\[
W^{VT} = \begin{pmatrix}
w_{11} & \cdots & w_{1J} \\
\vdots & \ddots & \vdots \\
w_{1J} & \cdots & w_{1J}
\end{pmatrix}
\]

where \( w_{ij} \) represents a basis vector of dimension \( RD \). Using
the bilinear model, the HMM mean vector \( y(s, n) \) can be
approximated by mixing the basis vectors with weights, which
is given by the tensor product of the speaker and noise
weight vectors:

\[
y(s, n) = a_i^TWb_n.
\]

Given \( Y \), the model parameters can be determined by mini-
mizing the reconstruction error:

\[
\min_{A,B,W} \|Y - (W^{VT}A)^{VT}B\|^2.
\]

Tenenbaum and Freeman [4] propose the following iterative
procedure for computing the model parameters:

1) Decompose \( Y = UDV^T \) by singular value decomposition
(SVD) and set \( B \) to be the first \( J \) rows of \( V^T \).

2) Decompose \( (YB)^{VT} = UDV^T \) and set \( A \) to be the first \( I \)
rows of \( V^T \).

3) Decompose \( (Y^{VT}A)^{VT} = UDV^T \) and set \( B \) to be the first \( J \)
rows of \( V^T \).

4) Iterate 2) and 3) until \( A \) and \( B \) converge. Upon con-
vergence, compute \( W \) using an equation in Eq. (5), e.g.,
\( W = [(YB)^{VT}A]^VT \). The basis matrix \( W \) is common across
speakers and noises and is used in the adaptation equation.

2.2 Adaptation to Speaker and Noise Using the Bilinear
Model

Using the basis matrix of the bilinear model from the pre-
vious section, we express the updated model for the tar-
gest speaker and noise as a product of the basis matrix and
two weight vectors, one each for speaker and noise, plus the
HMM mean vector of the clean speech SI model:

\[
\hat{\mu} = a_i^TWb + \mu_{SI}
\]
where \( \mathbf{a} \in \mathbb{R}^{D_s \times 1} \) and \( \mathbf{b} \in \mathbb{R}^{D_t \times 1} \) denote the speaker and noise weight vectors, respectively, which are estimated using adaptation data. The updated model can be depicted as in Fig. 1.

Given adaptation data \( \mathbf{O} = \{ \mathbf{o}_t, t = 1, \ldots, T \} \), the weight vectors are estimated in the maximum likelihood (ML) framework:

\[
\Lambda_{\text{ML}} = \arg \max_{\Lambda} p(\mathbf{O}|\Lambda)
\]

\[
= \arg \max_{\Lambda} \log p(\mathbf{O}|\Lambda)
\]

where \( \Lambda = \{ \mathbf{a}, \mathbf{b} \} \) denotes the set of parameters to be estimated. In finding the weight vectors using the expectation-maximization (EM) algorithm [17], the auxiliary function to be maximized is given by (discarding the terms that are independent of \( \mathbf{a} \) or \( \mathbf{b} \))

\[
Q(\mathbf{a}, \mathbf{b}) = -\frac{1}{2} \sum_{i=1}^{T} \sum_{r=1}^{R} \gamma_r(t)(\mathbf{o}_t - \mathbf{s}_r(\mathbf{a}, \mathbf{b}))\Sigma_r^{-1}(\mathbf{o}_t - \mathbf{s}_r(\mathbf{a}, \mathbf{b}))^T
\]

where \( \gamma_r(t) \) denotes the \textit{a posteriori} probability of occupying the \( r \)th mixture component at \( t \) given \( \mathbf{O}, \Sigma_r \) the covariance matrix for Gaussian mixture component \( r \) of the clean speech SI HMM (a diagonal matrix is used in our experiments), and

\[
\mathbf{s}_r(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{W}_r \mathbf{b} + \mu_{\text{St},r}
\]

where \( \mathbf{W}_r \in \mathbb{R}^{D_s \times J} \) and \( \mu_{\text{St},r} \) denote the matrix and vector from \( \mathbf{W} \) and \( \mu_{\text{St}} \), respectively, corresponding to the \( r \)th mixture component. We propose the following iterative procedure to determine \( \mathbf{a} \) and \( \mathbf{b} \):

1) Initialize \( \mathbf{a} \) with \( \tilde{\mathbf{a}} = (1/S) \sum_{s=1}^{S} \mathbf{a}_s \).

2) Set \( \mathbf{W}_a = \mathbf{a}^T \mathbf{W} \). Then, Eq. (13) becomes \( \mathbf{s}_r(\mathbf{b}) = \mathbf{W}_r \mathbf{b} + \mu_{\text{St},r} \). The noise weight vector that maximizes the likelihood can be obtained by setting \( \partial Q(\mathbf{b})/\partial \mathbf{b} = 0 \):

\[
\mathbf{b} = \left[ \sum_{r=1}^{R} \sum_{i=1}^{T} \gamma_r(t) \mathbf{W}_r^T \Sigma_r^{-1} \mathbf{W}_r \right]^{-1} \times \left[ \sum_{r=1}^{R} \sum_{i=1}^{T} \gamma_r(t) \mathbf{W}_r^T \Sigma_r^{-1}(\mathbf{o}_t - \mu_{\text{St},r}) \right]
\]

where \( \mathbf{W}_r \in \mathbb{R}^{D_s \times J} \) denotes the vector from \( \mathbf{W}_a \) that corresponds to the \( r \)th mixture component.

3) Set \( \mathbf{W}_b = \mathbf{W}_b \). Then, Eq. (13) becomes \( \mathbf{s}_r(\mathbf{a}) = \mathbf{a}^T \mathbf{W}_b + \mu_{\text{St},r} \). The speaker weight vector that maximizes the likelihood can be obtained by setting \( \partial Q(\mathbf{a})/\partial \mathbf{a} = 0 \):

\[
\mathbf{a} = \left( \sum_{r=1}^{R} \sum_{i=1}^{T} \gamma_r(t) \mathbf{W}_r^T \Sigma_r^{-1} \mathbf{W}_r \right)^{-1} \times \left[ \sum_{r=1}^{R} \sum_{i=1}^{T} \gamma_r(t) \mathbf{W}_r^T \Sigma_r^{-1}(\mathbf{o}_t - \mu_{\text{St},r}) \right]
\]

where \( \mathbf{W}_r \in \mathbb{R}^{D_t \times J} \) denotes the vector from \( \mathbf{W}_b \) that corresponds to the \( r \)th mixture component.

4) Iterate 2) and 3) until the weight vectors converge. Upon convergence, the updated model is given by

\[
\hat{\mathbf{a}} = \mathbf{b}^T \mathbf{W} \mathbf{b} + \mu_{\text{St}}
\]

where \( \mathbf{a} \) and \( \mathbf{b} \) are converged weight vectors. In the experiments, we stop the iteration when the following conditions are met:

\[
\frac{||\mathbf{a}_{i+1} - \mathbf{a}_i||}{||\mathbf{a}_i||} < \epsilon \text{ and } \frac{||\mathbf{b}_{i+1} - \mathbf{b}_i||}{||\mathbf{b}_i||} < \epsilon
\]

where \( \mathbf{a}_i \) and \( \mathbf{b}_i \) are current estimates of the weights and \( \mathbf{a}_{i+1} \) and \( \mathbf{b}_{i+1} \) are updated estimates of the weights. In our experiments, we used \( \epsilon = 0.01 \).

### 2.3 Adaptation Using Linear Models

The application of PCA to training acoustic models produces the linear model, so-called the eigenvoice model [2]. The decomposition of training models by PCA is straightforward. Given training models \( \{\mu(s, n)\} \), the sample covariance matrix is given by

\[
\mathbf{C} = \frac{1}{SN - 1} \sum_{s=1}^{S} \sum_{n=1}^{N} (\mu(s, n) - \bar{\mu})(\mu(s, n) - \bar{\mu})^T
\]

where \( \bar{\mu} = \frac{1}{SN} \sum_{s=1}^{S} \sum_{n=1}^{N} \mu(s, n) \).

Then, the \( K \) dominant eigenvectors of the sample covariance matrix become basis vectors. The model for the target speaker and noise is assumed to be lied in the space spanned by the basis vectors:

\[
\hat{\mathbf{u}} = \mathbf{\phi} \mathbf{x} + \bar{\mu}
\]

where \( \mathbf{\phi} = [\phi_1, \cdots, \phi_K] \) (\( \phi_k \) is an eigenvector) and \( \mathbf{x} \in \mathbb{R}^{K \times 1} \) \((K \leq SN - 1)\) is the weight vector to be estimated. In the ML framework, the weight vector is computed as

\[
\mathbf{x} = \left( \sum_{r=1}^{R} \sum_{i=1}^{T} \gamma_r(t) \mathbf{\phi}_r^T \Sigma_r^{-1} \mathbf{\phi}_r \right)^{-1} \times \left[ \sum_{r=1}^{R} \sum_{i=1}^{T} \gamma_r(t) \mathbf{\phi}_r^T \Sigma_r^{-1}(\mathbf{o}_t - \bar{\mu}_r) \right]
\]
where $\Phi_r \in \mathbb{R}^{D \times K}$ and $\mu_r \in \mathbb{R}^{D \times 1}$ denote the matrix and the vector from $\Phi$ and $\bar{\mu}$, respectively, corresponding to the $r$th mixture component.

3. Experiments

We used the AURORA4 corpus [18] in our experiments. The corpus contains the Wall Street Journal (WSJ) corpus WSJ0 [19] and its noisy versions. We used the speech which was sampled at 16 kHz. Some of the system setup is shown in Table 1. Cepstral mean normalization (CMN) was performed on training and test feature vectors.

In the training phase, we first built the clean speech SI HMM using the clean utterances of the AURORA4 corpus. The training speech consisted of 7,138 utterances of 83 speakers from the standard SI-84 training set of the WSJ0 corpus (the problematic data of one speaker were not used). Next, we built clean speech training models by applying MLLR adaptation [9] with 32 regression classes followed by maximum a posteriori (MAP) adaptation [20] to the clean speech SI HMM using clean utterances for each of the training speakers. Last, we built two versions of noisy speech training models using the clean speech of the AURORA4 corpus that was corrupted by the noises from the NOISEX-92 database [21]. For each of the training speakers, the ‘pink’ and ‘white’ noises from the NOISEX-92 database were added to the clean speech of the AURORA4 corpus at an SNR of 15 dB. We transformed the clean speech SI HMM by MLLR + MAP adaptation using the noisy utterances. These 249 adapted models (83 clean speech training models, 83 training models of the pink noise, and 83 training models of the white noise) were used to build the linear and bilinear models.

In adaptation tests, we used the clean and noisy test sets from the AURORA4 corpus, each containing utterances of eight test speakers. The noisy test set contained noisy utterances of six types of noise with SNRs varying between 15 and 5 dB having an average SNR of 10 dB: airports, babble, cars, restaurants, street traffic, and train terminals and stations. For each test noise condition and speaker, adaptation data with the lengths of 5 and 10 s were used. All adaptation was performed in a supervised mode. For each test noise condition, 290 utterances from the eight test speakers were recognized by updated models.

The clean speech SI HMM gives the word recognition accuracy of 67.51%, which is averaged over the seven test environments. Table 2 shows the recognition results of adapted models. The dimensions of the weight vectors reported in Table 2 are $K = 20$ in the linear model, and $(I = 20, J = 2)$ in the bilinear model. As shown in Table 2, the bilinear model outperforms the linear model in most test environments; the word recognition errors are reduced by 16.0% and 15.0% on average for adaptation data of 5 and 10 s in length, respectively, compared with the linear model. The results indicate the effectiveness of using separate weight vectors for speaker and noise in the bilinear approach. The improvement comes at the cost of additional computational load during the iterative procedure for computing the two weight vectors. In obtaining updated models using bilinear models, 5.9 and 5.6 iterations were needed on average until convergence for the adaptation of 5 s and 10 s, respectively. As a result, the ratios of the time taken to obtain updated models using bilinear models to the time taken to obtain updated models using linear models are 5.5 and 5.4 on average for the adaptation data of 5 s and 10 s, respectively. Nevertheless, the additional computational load can be tolerated if the adaptation is performed while a device is idle.

4. Conclusions

We presented the adaptation of acoustic models to the target speaker and noise environment for robust HMM-based speech recognition. Two factors that are present in the training models, speaker and noise, are decoupled to build the bilinear model. The bases from the decomposition are then used for the adaptation of HMM mean parameters to the target speaker and noise.

References

[1] L.R. Rabinder, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proc. IEEE, vol.77, no.2, pp.257–286, Feb. 1989.
[2] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, “Rapid speaker adaptation in eigenvoice space,” IEEE Trans. Speech Audio Process., vol.8, no.6, pp.695–707, Nov. 2000.
[3] I.T. Jolliffe, Principal Component Analysis, 2nd ed., Springer, New York, 2002.
[4] J.B. Tenenbaum and W.T. Freeman, “Separating style and content with bilinear models,” Neural Comput., vol.12, no.6, pp.1247–1283, June 2000.

[5] I. Mpiperis, S. Malassiotis, and M.G. Strintzis, “Bilinear models for 3-D face and facial expression recognition,” IEEE Trans. Inf. Forensic Secur., vol.3, no.3, pp.498–511, Sept. 2008.

[6] H.J. Song, Y. Jeong, and H.S. Kim, “A new method for speaker adaptation using bilinear model,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), pp.4365–4368, 2009.

[7] V. Popa, J. Nurminen, and M. Gabbouj, “A novel technique for voice conversion based on style and content decomposition with bilinear models,” Proc. Interspeech, pp.2655–2658, 2009.

[8] Y.-Q. Wang and M.J.F. Gales, “Speaker and noise factorisation on the AURORA4 task,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), pp.4584–4587, 2011.

[9] C.J. Leggetter and P.C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” Comput. Speech Lang., vol.9, no.2, pp.171–185, April 1995.

[10] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, “HMM adaptation using vector Taylor series for noisy speech recognition,” Proc. ICSLP, pp.869–872, 2000.

[11] M.L. Seltzer and A. Acero, “Separating speaker and environmental variability using factored transforms,” Proc. Interspeech, pp.1097–1100, 2011.

[12] V.V. Digalakis, D. Ritschew, and L.G. Neumeyer, “Speaker adaptation using constrained estimation of Gaussian mixtures,” IEEE Trans. Speech Audio Process., vol.3, no.5, pp.357–366, Sept. 1995.

[13] Y.-Q. Wang and M.J.F. Gales, “An explicit independence constraint for factorised adaptation in speech recognition,” Proc. Interspeech, pp.1233–1237, 2013.

[14] J. Latorre, V. Wan, M.J.F. Gales, L. Chen, K.K. Chin, K. Knill, and M. Akamine, “Speech factorization for HMM-TTS based on cluster adaptive training,” Proc. Interspeech, pp.971–974, 2012.

[15] O. Saz and T. Hain, “Asynchronous factorisation of speaker and background with feature transforms in speech recognition,” Proc. Interspeech, pp.1238–1242, 2013.

[16] Y. Jeong, “Acoustic model adaptation based on tensor analysis of training models,” IEEE Signal Process. Lett., vol.18, no.6, pp.347–350, June 2011.

[17] A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” J. R. Stat. Soc. Ser. B-Stat. Methodol., vol.39, no.1, pp.1–38, 1977.

[18] N. Parihar and J. Picone, “Aurora working group: DSR front end LVCSR evaluation AU/384/02,” Tech. Rep., Inst. for Signal and Information Process., Mississippi State University, 2002.

[19] D.B. Paul and J.M. Baker, “The design for the Wall Street Journal-based CSR corpus,” Proc. DARPA Speech and Natural Language Workshop, pp.357–362, 1992.

[20] J.L. Gauvain and C.-H. Lee, “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains,” IEEE Trans. Speech Audio Process., vol.2, no.2, pp.291–298, April 1994.

[21] A. Varga and H.J.M. Steeneken, “Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems,” Speech Commun., vol.12, no.3, pp.247–251, July 1993.