An Ensemble Speaker and Speaking Environment Modeling Approach to Robust Speech Recognition

Yu Tsao, *Member*, *IEEE*, and Chin-Hui Lee, *Fellow, IEEE*

Abstract—**We propose an ensemble speaker and speaking environment modeling (ESSEM) approach to characterizing environments in order to enhance performance robustness of automatic speech recognition systems under adverse conditions. The ESSEM process comprises two phases, the offline and the online. In the offline phase, we prepare an ensemble speaker and speaking environment space formed by a collection of super-vectors. Each super-vector consists of the entire set of means from all the Gaussian mixture components of a set of hidden Markov models that characterizes a particular environment. In the online phase, with the ensemble environment space prepared in the offline phase, we estimate the super-vector for a new testing environment based on a stochastic matching criterion. In this paper, we focus on methods for enhancing the construction and coverage of the environment space in the offline phase. We first demonstrate environment clustering and partitioning algorithms to structure the environment space well; then, we propose a minimum classification error training algorithm to enhance discrimination across environment super-vectors and therefore broaden the coverage of the ensemble environment space. We evaluate the proposed ESSEM framework on the Aurora2 connected digit recognition task. Experimental results verify that ESSEM provides clear improvement over a baseline system without environmental compensation. Moreover, the performance of ESSEM can be further enhanced by using well-structured environment spaces. Finally, we confirm that ESSEM gives the best overall performance with an environment space refined by an integration of all techniques.**

Index Terms—noise robustness, environment modeling

I. Introduction

THE performance of automatic speech recognition (ASR) systems has improved significantly since the hidden Markov model (HMM) was adopted as a fundamental tool to model speech signals in the mid-70s [1-3]. Deployment of ASR in mobile devices–such as personal data assistants and cell phones, as well as in client services such as online-ticketing systems, and customer care management systems in call centers–has greatly facilitated human-machine interfaces in recent years. However, the applicability of HMM-based ASR is limited due to one critical issue: data-driven HMM-trained speech models do not generalize well from training to testing conditions. Such an inevitable mismatch is generally derived from: 1) speaker effects, e.g., speech production, accent, dialect, and speaking rate differences; 2) speaking environment effects, e.g., interfering noise, transducers and transmission channel distortions. Although some functions can model particular distortion sources well, the form of an unknown combination of speaker and environment distortions is often unavailable or cannot be exactly specified.

The mismatch between training and testing conditions can be viewed in the signal, feature or model space, as illustrated in Fig. 1 [4, 5]. First, in the signal space, S_X and S_Y denote the speech signals in the training and testing conditions, respectively. We represent the distortion observed in the signal space as $D_S(.)$. A following feature extraction procedure converts the speech signals to a few compact and perceptually meaningful features. We represent training and testing features as F_X and F_Y in Fig. 1. From these features, the statistical models Λ_X and Λ_Y can then be trained. We denote the distortions observed in the feature and model-spaces as $D_F(.)$ and $D_M(.)$, respectively.

The approaches that tackle the mismatch problems can be roughly classified into three categories: C_1, C_2, and C_3 (Fig. 1). The first category of C_1 approaches is often referred to as speech enhancement methods because the objective is to produce robust features less sensitive to environment changes

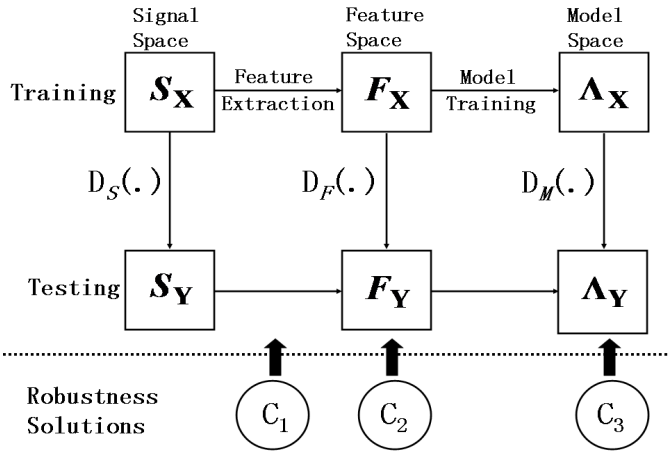


Fig. 1. Mismatch modeling and three classes of solutions.

Manuscript received Aug. 13, 2008. This work was supported by a Texas Instruments Leadership University (TILU) grant.

Yu Tsao and Chin-Hui Lee are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. 30332 USA (e-mail: {yutsao, chl}@ece.gatech.edu).

and reduce mismatch in the feature extraction stage. These approaches usually involve a new feature extraction procedure. One such C_1 approach, spectral subtraction (SS), and its extensions [6, 7], significantly reduce additive noise by subtracting the noise power spectrum from each speech frame. Likewise, cepstral mean subtraction (CMS) [8, 9] normalizes speech features in the cepstral domain by subtracting the means from speech frames. Techniques using second or higher order cepstral moment normalization adjust the distribution of noisy speech features closer to that of the clean ones and provide further improvement over the first order CMS [10-12]. More recently, the ETSI advanced front-end (AFE) is proposed to achieve good performance in ASR noise robustness [13]. This ETSI AFE removes mismatch by using several stages of noise reduction schemes, including a two-stage Wiener filter, signal noise ratio (SNR)-dependent waveform processing, cepstrum calculation, and blind equalization.

The second category of approaches removes mismatches in the feature-space; we denote them as C_2 in Fig. 1. These methods form a parametric function to model the distortion $D_F(.)$ between the training and testing speech features. The parametric function is estimated based on some optimality criterion and is used to compensate testing features. The codeword dependent cepstral normalization (CDCN) algorithm [14] and the stereo-based piecewise linear compensation environments (SPLICE) technique [15], for example, perform feature compensation with a correction vector, which is estimated or located with a VQ codeword that indicates the gap between the training and testing environments. Similarly, both feature-space maximum likelihood linear regression (fMLLR) [16] and feature-space eigen-MLLR [17] compute affine transformations to compensate noisy features based on a maximum likelihood (ML) criterion.

The third class of approaches, C_3, reduces mismatches by adjusting parameters in the acoustic models so that they can accurately match various adverse testing conditions. These approaches intend to map the original acoustic models, Λ_X, to a new set of acoustic models, Λ_Y, that matches the testing features. For these approaches, a set of speech segments from the testing environment is required for the mapping process, and these speech samples are called adaptation data. The model-mapping process can be done in either a direct or an indirect manner [18]. A direct mapping finds the target acoustic models for the unknown testing environment directly. When sufficient adaptation data is available, such direct mapping achieves a good performance. Maximum a posteriori (MAP) estimation [19] is a well-known method belonging to this category. On the other hand, indirect adaptation models the difference between training and testing conditions by a mapping function that transforms the original models, Λ_X, to new models, Λ_Y. The most often used form of the mapping function is an affine transformation. Maximum likelihood linear regression (MLLR) [20] and its Bayesian version, maximum a posteriori linear regression (MAPLR) [18, 21], have been adopted with good success, where the affine transformations are estimated through ML and MAP learning, respectively. Moreover, stochastic matching [4, 5] provides an effective way to estimate the compensation factor in a maximum likelihood self adaptation manner. Another mapping function is a distortion model that characterizes the mismatch between Λ_X and Λ_Y. A vector Taylor series (VTS) expansion is often used to approximate the distortion model. Examples include the joint compensation of additive and convolutive distortion (JAC) [22] and VTS-based HMM adaptation [23]. When comparing the direct and indirect adaptation approaches, the later ones are generally more effective when a small set of adaptation data is available. Therefore, extensions have been proposed to the direct mapping approaches. One successful extension is to introduce a hierarchical structure as a flexible parameter tying strategy in estimating HMM parameters. Structural MAP (SMAP) [24] uses such a hierarchical structure and shows performance improvements over the conventional MAP. Moreover, a unified framework for a joint MAP adaptation of transformation (indirect) and HMM (direct) parameters has been proposed [25] to not only achieve rapid model adaptation with limited adaptation data but also continuously enhance performance when a large set of adaptation data is available.

In this paper, we present an ensemble speaker and speaking environment modeling (ESSEM) approach to characterizing unknown environments. ESSEM models each environment of interest with a super-vector, consisting of the entire set of mean vectors from all the Gaussian components in the HMM set for that particular environment. A collection of such super-vectors obtained from many speaker and speaking conditions forms an environment space. With such an environment configuration, ESSEM estimates a target super-vector for an unknown testing environment online with a mapping function. The estimated super-vector is then used to construct HMMs for the testing environment. In contrast with multicondition training [26], which trains a set of models to cover a wide range of acoustic conditions collectively, the proposed ESSEM approach generates a new set of models that is more focused for a specific environment.

The rest of this paper is organized as follows. We first present a general framework of the ESSEM approach in Section II. Then, we detail the techniques to refining environment space in Section III. Section IV introduces the online super-vector estimation with the refined environment space. In Section V, we report the experimental results and discussion. Finally, we conclude our findings in Section VI.

II. The Ensemble Speaker And Speaking Environment Modeling Framework

Based on our previous study [27], we know that two classes of mapping procedures are applicable to find the target super-vector, namely direct and indirect ESSEM approaches. For direct ESSEM, we estimate the target super-vector through a mapping function along with a large collection of environment-specific super-vectors. For indirect ESSEM, we use a mapping function to estimate a transformation with another large collection of transformations, each corresponding to the mapping required for a particular known

environment over an anchor or reference super-vector. Then, we compute the target super-vector with the estimated transformation and the anchor super-vector [27]. Similar frameworks to indirect ESSEM show good performance in speaker adaptation [17, 28]. In this study, we limit our discussion on direct ESSEM.

The proposed ESSEM approach is derived from the stochastic matching framework [4, 5]. Therefore, we review the stochastic matching framework before introducing the ESSEM approach.

A. Stochastic Matching

First, we briefly review the ML-based stochastic matching framework. In speech recognition, we are interested in the following problem: given a set of trained acoustic models, Λ_X, and a set of testing data, F_Y, as in Fig. 1, we want to decode a word sequence $\boldsymbol{W}=\{W_1, W_2,\ldots, W_L\}$ such that:

$$\boldsymbol{W}' = \underset{\boldsymbol{W}}{argmax}\, P(F_Y \mid \boldsymbol{W}, \Lambda_X)\, P(\boldsymbol{W}) . \quad (1)$$

The stochastic matching approach uses a mapping function, $\mathbf{G}_\varphi$, with parameters φ to transform the original acoustic models, Λ_X, to a desired set of models, Λ_Y, for the testing environment by:

$$\Lambda_Y = \mathbf{G}_\varphi(\Lambda_X) . \quad (2)$$

The form of the mapping function depends on the amount of adaptation data and the type of acoustic mismatch. We call φ the nuisance parameters that are only used in the mapping procedure but not involved in the recognition procedure. From (1) and (2), we can formulate a joint maximization equation:

$$(\varphi', \boldsymbol{W}') = \underset{(\varphi, \boldsymbol{W})}{argmax} P(F_Y \mid \varphi, \boldsymbol{W}, \Lambda_X)\, P(\boldsymbol{W}) . \quad (3)$$

An iterative procedure can be used to solve (φ, $\boldsymbol{W}$). Since our main interest is to compute the parameters φ, we remove the dependence of $\boldsymbol{W}$ for notational simplicity and rewrite (3) as:

$$\varphi' = \underset{\varphi}{argmax} P(F_Y \mid \varphi, \Lambda_X) . \quad (4)$$

The nuisance parameters in (4) are estimated based on the expectation-maximization (EM) algorithm [29]. A simpler form (a bias vector stochastic matching [4]) and a more complex structure (a non-linear transformation [5]) of mapping structure have been used to model the mismatch factors successfully.

B. Ensemble Speaker and Speaking Environment Modeling

Next, we present the proposed ensemble speaker and speaking environment modeling (ESSEM) approach. Similar to stochastic matching, the final goal of ESSEM is to estimate a mapping function, $\mathbf{G}_\varphi$, so as to find a set of acoustic models for the testing environment. However, instead of using one set of models in (2), ESSEM prepares multiple sets of acoustic models for many different acoustic environments. We believe such an extension can effectively represent the complex structure of the environment space.

The ESSEM framework comprises two stages: offline and online phases. In the offline phase, we collect speech data from different speaker and speaking environments, e.g., different speakers, noise types, and channel distortions. A collection of data with many different combinations of adverse conditions from real-world environments is usually prohibitive. We address this issue by artificially simulating a wide range of speaker and speaking environment conditions [27]. The simulation process also enables us to quantitatively and qualitatively control the property and the coverage of the environment space. After collecting or simulating P sets of training data for P different speaker and speaking environments, we can train P sets of HMMs, Λ_p, p=1,…, P. The entire set of mean parameters within a set of HMMs is then concatenated into a super-vector, V_p, p=1,…,P. The order of concatenating the mean parameters is unified among all the super-vectors and is followed by a reference HMM set. If one set of HMMs contains M Gaussian mixture components, and every mean vector has D dimensions, then every super-vector has R (R=D×M) dimension. These P super-vectors form an ensemble speaker and speaking environment space, $\Omega_{\mathbf{V}} = \{V_1 V_2 \ldots V_P\}$, that serves as *a priori* knowledge for estimating the super-vector representing of the target condition. In the following, we call this environment space the ESS space for notational simplicity.

In the online phase, we estimate the target super-vector, V_Y, for a testing environment with the ESS space prepared in the offline phase:

$$V_Y = \mathbf{G}_\varphi(\Omega_{\mathbf{V}}) , \quad (5)$$

$$\varphi' = \underset{\varphi}{argmax} P(F_Y \mid \varphi, \Omega_{\mathbf{V}}) . \quad (6)$$

Similar to (4), we can use an EM algorithm, i.e. [29], to estimate the nuisance parameters φ in (6).

In this paper, we focus on the offline preparation stage and propose methods to enhance the environment configuration. As will be discussed later in the experiments section, an enhanced environment configuration can facilitate both accuracy and efficiency in operating conditions.

III. Offline Environment Space Preparation

In this section, we present techniques to enhance the environment configuration. To well structure the environment space, we develop environment clustering (EC) and environment partitioning (EP) algorithms. To increase the discrimination power of the environment structure, we derive two environment training algorithms based on minimum classification error (MCE) training [30, 31].

A. Structuring the Environment Spaces

The objective of environment clustering (EC) resembles that of well-known subset selection methods [32] that select a subset from the entire set of components to model a signal of interest. On the other hand, environment partitioning (EP) is similar to the piecewise-polynomial and spline functions [33] that approximate complicated functions with local polynomial representations.

1) Environment Clustering (EC)

First, we introduce EC to cluster the ensemble environments into several groups with each group consisted of environments having close acoustic properties; environments within a same group then form a sub-space. In this paper, we present a

hierarchical clustering procedure to construct a tree structure. The root of the tree is the entire set of training environments, and the tree is partitioned into several layers, with each layer of environment clustering performed based on similarity between each pair of environments. In the offline phase, the super-vectors belonging to a same cluster form an environment clustering (EC) ESS sub-space. For a hierarchical tree with C groups (including the root node, intermediate nodes, and leaf nodes), we can categorize the original ESS space in (5) into C sub-spaces: $\Omega_{\mathbf{V}} = \{\Omega_{\mathbf{V}^{(1)}} \cup \Omega_{\mathbf{V}^{(2)}} ... \cup \Omega_{\mathbf{V}^{(C)}}\}$. We specify a function, $R(.)$, to determine a representative super-vector for each of these sub-spaces; for example, the super-vector, $\mathrm{V}_{\mathrm{rep}}^{(c)}$, represents the c-th cluster, $\Omega_{\mathbf{V}^{(c)}}$:

$$\mathrm{V}_{\mathrm{rep}}^{(c)} = R(\Omega_{\mathbf{V}^{(c)}}) . \quad (7)$$

More details about the establishment of a hierarchical tree and the calculation of the function $R(.)$ can be found in our previous study [31]. The similarity measure between a pair of environments can be defined either by a deterministic distance between their corresponding super-vectors or based on knowledge about the acoustic difference between them. Using a deterministic distance allows us to construct a hierarchical tree in a data-driven manner, while it is perceptually meaningful to use the acoustic knowledge as the similarity measure. For example, when we obtain super-vectors from many different acoustic environments, we can form speaker sub-space, $\Omega_{\mathbf{V}^{(\rho)}}$, noise sub-space, $\Omega_{\mathbf{V}^{(b)}}$, and channel sub-space, $\Omega_{\mathbf{V}^{(h)}}$:

$$\Omega_{\mathbf{V}} = \{\Omega_{\mathbf{V}^{(\rho)}} \cup \Omega_{\mathbf{V}^{(b)}} \cup \Omega_{\mathbf{V}^{(h)}}\}. \quad (8)$$

A combination of the deterministic distance and acoustic knowledge can be another tree construction scheme. For such a case, we first cluster environments based on the distortion sources they contain; then, we build a hierarchical tree for each distortion domain based on some deterministic distances.

2) Environment Partitioning (EP)

Next, we introduce the EP algorithm to structuring the ESS space. Instead of clustering environments, EP partitions each super-vector into several sub-vectors. Then, we collect each set of sub-vectors among all the training environments to form a sub-space. From our previous study [34], two types of super-vector partitioning are successful, namely, the mixture-based and feature-based EP techniques.

For mixture-based EP, we establish a tying structure to cluster Gaussian mixture components, as in the tree structure in SMAP [24], and thereby the entire set of Gaussian mixture components in a set of HMMs is classified into S clusters. We can use Mahalanobis, Bhattacharyya, or the divergence distance [35] to measure the similarity between a pair of Gaussian mixture components. Then, the original super-vector is partitioned into S sets of sub-vectors ($\mathrm{V}_p = [\mathrm{V}_{p,1}^{\mathrm{T}}, \mathrm{V}_{p,2}^{\mathrm{T}}, ..., \mathrm{V}_{p,S}^{\mathrm{T}}]^{\mathrm{T}}$, for the p-th super-vector). Each set of such sub-vectors from the P environments then forms a sub-space individually, $\Omega_{\mathbf{V}_s} = \{\mathrm{V}_{1,s}\, \mathrm{V}_{2,s} ... \mathrm{V}_{P,s}\}$, s=1,2,…S. Another tying method is to classify models with whole-word, or sub-word units, and accordingly their Gaussian mixture components, into different clusters based on acoustic or linguistic knowledge [34].

For feature-based EP, we tie different vector components, e.g., energy, static, first and second order time derivative coefficients. When tying coefficients into Z groups, the original super-vector is partitioned into Z sub-vectors ($\mathrm{V}_p = [\mathrm{V}_{p,1}^{\mathrm{T}}, \mathrm{V}_{p,2}^{\mathrm{T}}, ..., \mathrm{V}_{p,Z}^{\mathrm{T}}]^{\mathrm{T}}$, for the p-th super-vector). Then, we can construct Z sets of sub-spaces, $\Omega_{\mathbf{V}_z} = \{\mathrm{V}_{1,z}\, \mathrm{V}_{2,z} ... \mathrm{V}_{P,z}\}$, z= 1,2,…Z, with each sub-space spanned by a particular group of coefficients.

B. Increasing Coverage of the Environment Spaces

Traditionally, discriminative training methods, such as minimum classification error (MCE) [30], maximum mutual information estimation (MMIE) [36], minimum word/phone error (MWE/MPE) [37], and soft margin estimation (SME) [38], were used to refine accuracy of acoustic modeling. In the ESSEM framework, we use the discriminative training to maximize the separation between super-vectors in order to spread the coverage of the ESS space. Among these discriminative training methods, we adopt MCE training [30, 31] because its misclassification measure can represent a probabilistic distance between two classes. We propose two modes of training on the ESS space, intra-environment (intraEnv) and inter-environment (interEnv) training. For both intraEnv and interEnv training, the parameters in the ESS spaces are first estimated with the ML criterion; then refined by MCE training.

1) Intra-Environment (intraEnv) Training

We use intraEnv training to increase the separation between components in one particular environment. With the training data F_p of U_p utterances from the p-th environment, we have the objective function:

$$L(\mathrm{V}_p) = \frac{1}{U_p} \sum_{u=1}^{U_p} \frac{1}{1 + exp(-\gamma\, d(F_p^u, \mathrm{V}_p, \Psi) + \theta)} , \quad (9)$$

where F_p^u is the u-th training utterance for the p-th environment; both γ and θ are control parameters for the sigmoid function; Ψ represent the parameters other than means in HMMs. Since our goal is to minimize the objective function by adjusting parameters in the ESS space, Ψ is fixed across different environments. The misclassification measure $d(.)$ is defined as [30]:

$$d(F_p^u, \mathrm{V}_p, \Psi) = -\tilde{g}(F_p^u, \mathrm{V}_p, \Psi, W_c) + \tilde{\mathrm{G}}(F_p^u, \mathrm{V}_p, \Psi) , \quad (10)$$

$$\tilde{\mathrm{G}}(F_p^u, \mathrm{V}_p, \Psi) = \frac{1}{\eta} log\{\frac{1}{N} \sum_{n=1}^{N} exp[\eta \times \tilde{g}(F_p^u, \mathrm{V}_p, \Psi, W_n)]\} , \quad (11)$$

where η is a positive control parameter, W_c is the given correct transcription, and $\{W_1,…,W_N\}$ are the N-best decoded competing word sequences. The N-best are generated by decoding F_p^u using the HMMs for the p-th environment. We used a log-likelihood for the discrimination function, $\tilde{g}(.)$ in (10) and (11), and adopted the generalized probabilistic descent (GPD) algorithm [39] to update parameters in V_p iteratively:

$$\mathrm{V}_p(t+1) = \mathrm{V}_p(t) - \kappa \nabla L(\mathrm{V}_p)|_{\mathrm{V}_p = \mathrm{V}_p(t)} , \quad (12)$$

where κ is a step size.

2) Inter-Environment (interEnv) Training

For the interEnv training mode, we consider each environment, accordingly its super-vector, as a particular class in the ESS space. Then, we collect speech data $F_{train} = \{F_1,...,F_P\}$ of a total of U utterances for P different environments and define an objective function:

$$L(\Omega_{\mathbf{V}}) = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{1+exp(-\gamma\, d(F_{train}^{u},\Omega_{\mathbf{V}},\Psi)+\theta)}, \quad (13)$$

where F_{train}^{u} is the u-th utterance in the training set. The misclassification measure $d(.)$ is now defined as:

$$d(F_{train}^{u},\Omega_{\mathbf{V}},\Psi) = -\tilde{g}(F_{train}^{u},\Omega_{\mathbf{V}},\Psi,W_c) + \tilde{\mathrm{G}}(F_{train}^{u},\Omega_{\mathbf{V}},\Psi), \quad (14)$$

$$\tilde{\mathrm{G}}(F_{train}^{u},\Omega_{\mathrm{V}},\Psi) = \frac{1}{\eta}log\{\frac{1}{N}\sum_{n=1}^{N}exp[\,\eta\times\tilde{g}(F_{train}^{u},\Omega_{\mathrm{V}},\Psi,W_n)\,]\,\} \quad (15)$$

W_c is again the given correct transcription. $\{W_1,\ldots,W_N\}$ are the N-best decoded competing word sequences. In the optimization process, we know the target environment to any segment of the training data, and we generate W_n by using the HMMs for the n-th most competitive environment to that target environment. Parameters in the ESS space are then updated iteratively based on:

$$\Omega_{\mathbf{V}}(t+1) = \Omega_{\mathbf{V}}(t) - \kappa\nabla L(\Omega_{\mathbf{V}})|_{\Omega_{\mathrm{V}}=\Omega_{\mathrm{V}}(t)}. \quad (16)$$

After performing intraEnv and interEnv training for several iterations, we obtain MCE-refined ESS spaces. Fig. 2 illustrates the ESS spaces with ML, intraEnv, and intraEnv followed by interEnv training, respectively. Experimental results corresponding to the three panels will be presented in Section 5.2.

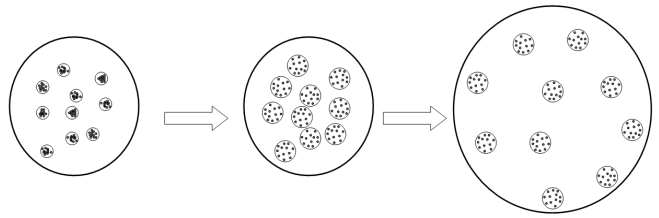


Fig. 2. IntraEnv and interEnv training to increase environment space discrimination.

IV. Online Target Super-vector Estimation

In this section, we introduce super-vector estimation in the online phase with the refined ESS spaces described in the previous section.

A. Environment Clustering

For the EC algorithm, we first conduct an online cluster selection to locate the most relevant cluster, $\Omega_{\mathbf{V}^{(c)}}$, whose representative super-vector has the highest likelihood to the testing data, F_Y:

$$\Omega_{\mathbf{V}^{(c)}} = arg\max_{\tilde{c}} P(F_{\mathrm{Y}} \mid R(\Omega_{\mathbf{V}^{(\tilde{c})}})). \quad (17)$$

With the selected cluster, $\Omega_{\mathbf{V}^{(c)}}$, and based on (5), we estimate the target super-vector, $\mathrm{V_Y}$, through:

$$\mathrm{V_Y} = \mathbf{G}_{\varphi}(\Omega_{\mathbf{V}^{(c)}}). \quad (18)$$

The structure defined by acoustic difference in (8) is advantageous when the testing condition is contaminated by a single distortion source, and the type of that distortion is determinable. For example, to remove channel distortions in a telephony service, we simply select the channel environment sub-space $\Omega_{\mathbf{V}^{(h)}}$ and estimate the target super-vector through $\mathrm{V_Y} = \mathbf{G}_{\varphi}(\Omega_{\mathbf{V}^{(h)}})$. Previous studies have verified such distortion-specific compensation operating well in speaker variations [40, 41], channel mismatches [42], and additive background noise [43]. It is also useful if each distortion has individually refined structure, such as hierarchical structure [44], to facilitate estimating the target super-vector in (18).

B. Environment Partitioning

For the mixture-bases EP technique, we estimate sub-vectors in $\mathrm{V_Y}$ through stochastic matching as shown in (5), individually. For example, the s-th sub-vector is estimated by:

$$\mathrm{V_{Y,}}_s = \mathbf{G}_{\varphi_s}(\Omega_{\mathbf{V}_s}),\ s=1, 2\ldots S. \quad (19)$$

Then, the target super-vector is formed by S sets of estimated sub-vectors:

$$\mathrm{V_Y} = [\,\mathrm{V_{Y,1}^T}, \mathrm{V_{Y,2}^T},\ldots,\mathrm{V_{Y,}}_S^{\mathrm{T}}]^{\mathrm{T}}. \quad (20)$$

For mixture-based EP, a same tying structure may not be well shared among different environments, especially for those environments with way different acoustic characteristics. We can handle this issue by sharing a same tying structure among environments with close acoustic properties. Therefore, it is favorable to combine EC when conducting mixture-based EP. On the other hand, feature-based EP estimates sub-vectors consisting different groups of coefficients individually. For the z-th group of coefficients, we have:

$$\mathrm{V_{Y,}}_z = \mathbf{G}_{\varphi_z}(\Omega_{\mathbf{V}_z}),\ \mathrm{z}=1, 2\ldots \mathrm{Z}. \quad (21)$$

Finally, we have the target super-vector by concatenating the Z sub-vectors into one super-vector:

$$\mathrm{V_Y} = [\,\mathrm{V_{Y,1}^T}, \mathrm{V_{Y,2}^T},\ldots,\mathrm{V_{Y,}}Z^{\mathrm{T}}]^{\mathrm{T}}. \quad (22)$$

V. Experimental Setup and Results

We evaluated the ESSEM framework on the Aurora2 database [45]. The multicondition training set was used to train HMMs and to build the ESS spaces. The training set includes 17 different speaking environments that are originated form the same four types of noise as in test Set A, at four SNR levels: 5dB, 10dB, 15dB, and 20dB, along with clean condition. We further divided the training set into two gender-specific subsets and obtained 34 speaker and speaking environments. We tested recognition on the complete evaluation set that consists of 70 testing conditions with 1001 utterances in each condition. A self-adaption (unsupervised) mode is used, and each testing utterance was first decoded into an N-best list and used for ESSEM model adaptation. We studied many online mapping functions, such as best first, linear combination, linear combination with a correction bias, and multiple cluster matching [27, 46]; in this paper, we selected the linear combination function throughout the following experiments.

This mapping function is also used in cluster adaptive training (CAT) [47] and eigenvoice [40]. With the EC algorithm, the online super-vector estimation in (18) becomes:

$$V_Y = \sum_{p=1}^{P^{(c)}} \hat{w}_p V_p , \quad (23)$$

where $\hat{w}_p$ is the p-th weighting coefficient in the linear combination function, and $P^{(c)}$ is the number of bases in the c-th sub-space. The set of weighting coefficients is estimated according to the ML algorithm:

$$\{\hat{w}_p\}_{p=1}^{P^{(c)}} = arg \max_{\{w_p\}_{p=1}^{P^{(c)}}} P(F_Y | \sum_{p=1}^{P^{(c)}} w_p V_p) . \quad (24)$$

As mentioned earlier, the online process of EC resembles the subset selection methods [32]. When comparing to CAT and eigenvoice, the major advantage of EC is to use the regional prior knowledge of the ESS space from the EC tree structure. This regional knowledge is critical to dealing with unknown testing conditions. With such regional knowledge, EC only uses the located group of super-vectors (the c-th cluster) to estimate the target super-vector in (24). Moreover, EC locates a representative HMM set through cluster selection. Instead of using the environment-independent HMM set, we use the located HMM set to calculate statistics needed in estimating the weighting coefficients in (24). The representative HMM set can provide more accurate statistics estimation than the environment-independent HMM set.

A. Environment Clustering and Environment Partitioning

We first evaluated the performance of the EC and EP algorithms. For this set of experiments, each speech frame was characterized by 39 coefficients consisted of 13 MFCCs with their first and second order time derivatives. An utterance-level CMS [8, 9] was performed for normalization. All digits were modeled by 16-state whole word HMMs with each state characterized by three Gaussian components. The silence and the short pause were modeled by three and one state, respectively, with each state characterized by six Gaussian mixture components. We tested ESSEM on gender independent (GI) and gender dependent (GD) systems. For the GI system, a GI HMM set was trained on the multicondition training data, and 34 environment-specific HMM sets were obtained by adapting (we used MAP [19]) mean vectors from the GI HMM set to particular environments. Next, we collected the mean vectors for these 34 HMM sets to build an ESS space. For the GD system, two GD HMM sets were first trained. Then, 17 environment-specific HMM sets for each gender were obtained by adapting mean vectors from that GD HMM set. Therefore, two ESS spaces corresponding to the two genders were prepared. An additional pair of HMM sets was prepared for automatic gender identification (AGI). For the AGI HMMs, each gender was modeled with 16 active states with each state characterized by 88 Gaussian mixture components.

1) EC on Gender Independent System

For the EC algorithm, we built a two-layer binary tree to cluster the 34 environments into seven groups (one root, two intermediate, and four leaf nodes). We clustered environments in a data-driven manner and observed that in the first layer of the 34 environments were exactly divided into two groups, each corresponding to one of the two genders. In the second layer, another two groups were classified roughly according to high/low SNR levels. In a preliminary experiment, we noted that if each group has too few super-vectors, the overall performance will drop. To maintain an adequate number of environments in every group, some environments were shared across different groups. These environments were located at the periphery of each group and usually at medium SNR levels. Finally, each cluster comprised 12 to 14 different environments in the second layer. For each node, we used the entire set of training data corresponding to that cluster of environments to train a representative super-vector $V_{rep}^{(c)}$ as shown in (7). The same topology used to train the environment-specific HMM sets was used to train the representative HMM set. We also tested using other methods to obtain the representative super-vector, $V_{rep}^{(c)}$, but using the full training set to train a new representative super-vector gave the best performance.

First, we compared ESSEM with the entire ESS space, EC with one-layer, and EC with two-layer tree structures and listed their results in Table I. For the Aurora2 evaluation, we are more interested in the results from SNR 0dB to 20dB conditions. Therefore, we only presented the average WERs over SNR 0dB to 20dB conditions for the three testing sets (Set A, Set B and Set C). Baseline is also listed as "GI-Baseline" in Table I for comparison. We denoted "GI-Full" for ESSEM with entire ESS space (P=34), "GI-EC(1)" for the EC-structured ESS space with a one-layer (cluster number C=3) tree, and "GI-EC(2)" for the EC-structured ESS space with a two-layer (cluster number C=7) tree. From Table I, we can see that "GI-EC(2)" achieves better performance than "GI-Full" and "GI-EC(1)" in all the three testing sets and gives a 19.98% (11.01% to 8.81%) WER reduction over "GI-Baseline".

TABLE I
AVERAGE WERS (IN %) FROM 0DB TO 20DB

Test conditions	Set A	Set B	Set C	Overall
GI-Baseline	11.27	10.76	11.00	11.01
GI-Full	9.16	9.41	9.10	9.25
GI-EC(1)	8.78	9.10	8.78	8.91
GI-EC(2)	8.72	9.07	8.48	8.81

We also listed the average WER for each SNR condition in Table II. In addition to WER, a statistical hypothesis test is usually used to verify whether a method is significantly better than another one. In this paper, we adopt the dependent t-Test (for matched-pair samples) for the hypothesis test [48, 49]. The dependent t-Test is especially suitable for the Aurora2 evaluation because: 1) each testing condition has a large amount of testing data (1001 utterances, more than 3000 words), so the measure of the average WER is reliable; 2) two methods have matched-pair sequences of samples, so a pair-wised testing of two methods is reasonable. For the dependent t-Test, we consider H_0 as "method two is not better than method one", and H_1 as "method two is better than method one". In Aurora2, each SNR condition has 10 results (10 pair-wised samples for t-Test). For hypothesis testing, P-values

can provide detailed information for experimental results [48, 49]. Thus in Table II, we list the corresponding P-values for all SNR conditions.

From Table II, we first find that EC(2) achieves lower WER than EC(1) in every SNR condition. Next, we observed that the P-values are 0.013 and 0.051, respectively, for 20dB and 0dB conditions. The small P-values imply consistent improvements of EC(2) over EC(1). We thus claim EC(2) is better than EC(1) in these two conditions. However, for 15dB, 10dB, and 5dB conditions, although WERs are reduced, the corresponding P-values are relatively large. These observations verify by further using a high/low SNR layer in building the tree structure for EC, ESSEM can better model the very low SNR or very high SNR conditions, while improvements of the medium SNR conditions may not be prominent.

TABLE II
WER(%) AND P-VALUE FOR TWO EC ESS SPACES

dB	WER		P-value
	GI-EC(1)	GI-EC(2)	EC(2) vs. EC(1)
20	1.61	1.58	0.013
15	2.06	2.04	0.308
10	3.54	3.52	0.341
5	8.74	8.68	0.105
0	28.58	28.24	0.051

2) EC on Gender Dependent System

In the GD system, we used every incoming testing utterance to: 1) determine speaker's gender; 2) select a GD HMM set and its corresponding ESS space; 3) perform ESSEM in an unsupervised self-adaptation manner; 4) test recognition with the ESSEM-adapted acoustic models. Similar to the GI system, we compared one-layer and two-layer tree structures and listed their results as "GD-EC(1)" and "GD-EC(2)", respectively, in Table III. Since the gender identity was determined by the AGI unit beforehand, the EC algorithm did not need an online cluster selection process as shown in (17) for the one-layer structure. To have a fair comparison, we used the AGI process followed by a speaking environment cluster selection as (17) to locate a representative HMM set. Then, we directly used the located HMM set to test recognition for the baseline and denoted the results as "GD-Baseline" in Table III. In Table IV, we list the detailed WERs and P-values of "GD-EC(2)" versus "GD-EC(1)" for all SNR conditions.

We observe similar results to the GI system. From Table III, "GD-EC(1)" and "GD-EC(2)" provide 7.88% (8.63% to 7.95%) and 8.57% (8.63% to 7.89%) WER reductions, respectively, over "GD-Baseline" in the overall performance. Next, by comparing "GD-EC(1)" and "GD-EC(2)" in Table IV, "GD-EC(2)" provides better performance almost in every SNR condition, and the improvement under very high SNR (20dB) and low SNR (0dB, 5dB) conditions are more significant.

TABLE III AVERAGE WERS (IN %) FROM 0DB TO 20DB

Test conditions	Set A	Set B	Set C	Overall
GD-Baseline	8.88	8.46	8.47	8.63
GD-EC(1)	8.10	7.89	7.79	7.95
GD-EC(2)	8.03	7.86	7.65	7.89

TABLE IV
WER(%) AND P-VALUE FOR TWO EC ESS SPACES

dB	WER		P-value
	GD-EC(1)	GD-EC(2)	EC(2) vs. EC(1)
20	1.16	1.14	0.075
15	1.62	1.62	0.422
10	2.86	2.83	0.262
5	7.59	7.49	0.048
0	26.55	26.34	0.037

3) EC+EP on Gender Dependent System

Finally, we present the ESSEM performance with EC followed by EP structuring on the ESS spaces. We used the same two-layer hierarchical tree structure for EC in the previous section followed by mixture-based and feature-based EP techniques. Again, we use a linear combination function for the online super-vector estimation. For mixture-based EP, the online estimation in (19) becomes:

$$\mathrm{V}_{\mathrm{Y},s} = \sum_{p=1}^{P^{(c)}} \hat{w}_p \mathrm{V}_{p,s} \ , s=1, 2\ldots S, \tag{25}$$

and for feature-based EP, the online estimation in (21) is:

$$\mathrm{V}_{\mathrm{Y},z} = \sum_{p=1}^{P^{(c)}} \hat{w}_p \mathrm{V}_{p,z} \ , z=1, 2\ldots Z, \tag{26}$$

where $P^{(c)}$ is the total number of bases in the selected c-th sub-space in the EC-structured ESS space.

The result to be compared with is the first stage EC algorithm alone as "GD-EC(2)" in Table III. The two types of two-stage structured ESS spaces were reported in Table V as "GD-EC(2)+EP(M)" and "GD-EC(2)+EP(F)" for using mixture-based and feature-based EP in the second stage, respectively. For mixture-based EP, we compared recognition performances using different clustering techniques. Among them, a hierarchical tree structure clustering method as suggested in [50] achieved the best performance. When using a hierarchical tree in mixture-based EP, we first constructed a tree structure based on a set of reference HMMs. In our implementation, we used the representative HMM set in each node from the EC stage to build the EP hierarchical tree. Each node in the EP tree structure, from the root node to leaf nodes, included a group of Gaussian mixture components. We built the EP tree by using a top-down k-means clustering algorithm and the Mahalanobis distance as a distance measure between Gaussians. In the offline phase, the original ESS spaces were partitioned into several sub-spaces by following these EP hierarchical trees. In the online phase, a searching process was conducted beforehand to find a node with a sufficient amount of adaptation statistics from leaf nodes to root node in the EP tree structure. Therefore, the total number of sub-vectors S in (25) was not predefined but determined based on the amount of adaptation data. For feature-based EP, we segmented each super-vector into three sub-vectors for three different types of coefficient components, namely, 13 static, 13 first and 13 second order time derivatives MFCCs. Then, we built three sub-spaces for three types of components. By comparing "GD-EC(2)" in Table III with "GD-EC(2)+EP(M)" and "GD -EC(2)+EP(F)" in Table V, we confirm that both the two types

of EP techniques can produce better overall performance than the one-stage EC algorithm alone.

Table VI lists WERs of "GD-EC(2)+EP(M)" and "GD -EC(2)+EP(F)" and P-values of them versus "GD-EC(2)". From Table IV and Table VI, we can see that the further improvements of EP(M) and EP(F) mainly come from SNR 0dB condition; both improvements at SNR=0dB are prominent (P-values=0.046 and 0.022, respectively). We also noted that in some conditions, "GD-EC(2)" achieves lower WERs than "GD -EC(2)+EP(M)" and "GD-EC(2)+EP(F)". For such cases, we estimate the P-values by hypothesizing "GD-EC(2)" is better than "GD-EC(2)+EP(M)" or "GD-EC(2)+EP(F)". For example at SNR=15dB, the P-value of "GD-EC(2)+EP(M)" versus "GD-EC(2)" is 0.287. With such a large P-value, we can not claim that "GD-EC(2)+EP(M)" is worse than "GD-EC(2)" even though "GD-EC(2)" provides lower WER.

TABLE V
AVERAGE WERS (IN %) FROM 0DB TO 20DB

Test conditions	Set A	Set B	Set C	Overall
GD-EC(2)+EP(M)	8.01	7.85	7.62	7.87
GD-EC(2)+EP(F)	7.99	7.79	7.63	7.84

TABLE VI
WER(%) AND P-VALUE FOR TWO EP ESS SPACES

dB	WER	P-value	WER	P-value
	GD-EC(2)+ EP(M)	vs. GD-EC(2)	GD-EC(2)+ EP(F)	vs. GD-EC(2)
20	1.14	0.453	1.14	0.339
15	1.63	0.287	1.64	0.247
10	2.83	0.411	2.83	0.470
5	7.50	0.323	7.49	0.432
0	26.22	0.046	26.10	0.022

B. MCE-refined Super-vectors in the ESS Space

In this section, we present ESSEM using ESS spaces with and without MCE training. We followed the procedure of Fig. 2 and conducted two sets of MCE training experiments: 1) intraEnv training, and 2) intraEnv followed by interEnv training. We applied the EC algorithm with a two-layer tree-structure to structure the ESS spaces and used a modified ETSI advanced front-end (AFE) suggested in [51] for feature extraction. Every feature vector consisted of 13 static plus their first and second order time derivatives. We followed a complex back-end model topology suggested in [13] to train HMMs where each digit was modeled by 20 mixtures per state and the silence and short pause were characterized by 36 mixtures per state. Again, we evaluated performance in both GI and GD systems. Based on an additional set of experiments, we observed that it is not necessary to have very complex HMM sets to identify speaker's gender as presented in the previous section. In this set of experiments, we simply used the same topology to other environment-specific HMMs to build a pair of gender-specific HMM sets for AGI.

1) MCE on Gender Independent System

Table VII lists the GI system results. "GI-Baseline" is baseline using the GI HMMs only [51]. "GI-ML", "GI-intraEnv", and "GI-intraEnv+interEnv" represent ESSEM using the ESS spaces trained by ML, intraEnv, and intraEnv followed by interEnv training, respectively. These results correspond to the left, middle, and right panels in Fig. 2. Table VIII lists the average WERs of "GI-ML" and "GI-intraEnv+interEnv" and P-values of "GI-intraEnv+interEnv" versus "GI-ML" in every SNR condition.

From Table VII, it is clear that ESSEM with the original ML-trained ESS space already achieved better performance of 12.85% (6.46% to 5.63%) relative WER reduction over "GI -Baseline". By comparing "GI-intraEnv", and "GI-ML", we observed that "GI-intraEnv" produces better performance than "GI-ML". Therefore, we verify that intraEnv training can refine ESS spaces and enhance overall ESSEM performance. By comparing "GI-intraEnv" and "GI-intraEnv+interEnv", we confirm that intraEnv followed by interEnv training provides further improvements over intraEnv training alone. Finally from Table VIII, we confirm that with intraEnv and interEnv training, ESSEM provides better performance for SNR 20dB to 5 dB conditions, while no significant improvement is achieved for SNR 0dB condition.

TABLE VII
AVERAGE WERS (IN %) FROM 0DB TO 20DB

Test conditions	Set A	Set B	Set C	Overall
GI-Baseline	5.92	6.69	7.11	6.46
GI-ML	5.12	6.07	5.78	5.63
GI-intraEnv	4.94	5.61	5.83	5.39
GI-intraEnv+interEnv	4.93	5.58	5.83	5.37

TABLE VIII
WER(%) AND P-VALUE FOR ML AND MCE TRAININGS

dB	WER		P-value
	ML	intraEnv+interEnv	intraEnv+interEnv vs. ML
20	0.62	0.51	0.006
15	1.04	0.85	0.001
10	2.33	2.04	0.001
5	6.05	5.47	0.006
0	18.10	17.95	0.338

2) MCE on Gender Dependent System

Next, we demonstrated the ESSEM results on the GD system. Table IX lists ESSEM results with ML-trained and MCE-trained ESS spaces as "GD-ML" and "GD-intraEnv +interEnv", respectively. We followed the procedure in Section 5.1.2 to obtain the baseline and denoted it as "GD-Baseline". We also list the average WERs and P-values in Table X. Similar to the GI case, ESSEM with an MCE-trained ESS space is better than that with an ML-trained ESS space. From Table X, we observed that after interEnv and intraEnv training, ESSEM is clearly improved among SNR 20dB to 5dB, while an insignificant degradation is shown in SNR 0dB (P-value= 0.484). From Table VII and Table IX, we can see the major improvements come from Set B, which consists of conditions that are not involved in the training set. This observation supports our claim that by using MCE, the coverage of the ESS space is broadened, and performance can be improved, especially for conditions under unseen noise types. However,

no improvement is achieved in testing Set C, where an additional channel distortion is involved. Moreover, from Table VIII and Table X, the MCE training gives no improvement for SNR 0dB condition. This should be a limitation of MCE training that aims at increasing distance among modeling units only according to the available training data. To cover very different testing conditions, we may need to include more environments in the training set or use a more complex online mapping function [46].

TABLE IX
AVERAGE WERS (IN %) FROM 0DB TO 20DB

Test conditions	Set A	Set B	Set C	Overall
GD-Baseline	5.11	5.38	6.56	5.51
GD-ML	4.72	5.21	5.60	5.09
GD-intraEnv+interEnv	4.64	4.99	5.64	4.98

TABLE X
WER(%) AND P-VALUE FOR ML AND MCE TRAININGS

dB	WER		P-value
	ML	intraEnv+interEnv	intraEnv+interEnv vs. ML
20	0.53	0.47	0.002
15	0.81	0.76	0.008
10	1.95	1.79	0.003
5	5.26	4.98	0.009
0	16.91	16.92	0.484

3) ESS Space Analysis: Before and After MCE Training

In addition to recognition results, we used two other measurements−1) separation of parameters in one HMM set for a particular environment; 2) difference between two HMM sets for two different environments−to investigate the discrimination properties of the ESS spaces. We used the first and second measurements to examine the intraEnv and interEnv training, respectively. The first measurement adopts the divergence distance to estimate each pair of Gaussian components and calculate an accumulated measurement for one particular environment. Details about this measurement can be found in [38]. We used the generalized log-likelihood ratio (GLLR) plot as the second measurement. For the GLLR plot, we first defined a target environment and its cohort environments. Then, we plotted histograms of GLLR scores with speech samples from the target and the cohort environments. More information about the GLLR plots can be found in our previous study [52]. After investigating, we found similar results for these two measurements for all the training environments. Therefore, we only presented the measurement results of−"subway noise, SNR=10dB, male speakers, from the GI system"−as the target environment.

Table XI shows the first measurement of before and after intraEnv training (the left versus the middle panels in Fig. 2). From Table XI, it is clear that after intraEnv training, the separation between parameters within one set of HMMs is increased. Next, we present the second measurement of before and after interEnv training (the middle versus the right panels in Fig. 2) in Fig. 3. To have a clear comparison, we used a Gaussian distribution to approximate the histograms and indicated the mean of each GLLR distribution in Fig. 3. From Fig. 3, we verify that the distance between the target and its competing environments is increased after interEnv training. Along with the improvements presented in Table VII and IX, we conclude that with discriminative training, we can enhance the discrimination power of the ESS space and thereby enable ESSEM to achieve better performance.

TABLE XI
DISTANCE IN AN ENVIRONMENT-SPECIFIC MODEL

Test HMMs	Before intraEnv	After intraEnv
Distance	72.80	74.44

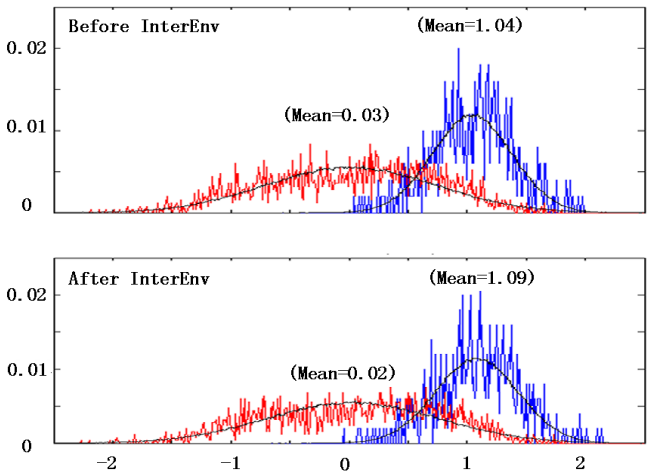


Fig. 3. Separation between environment-specific models.

4) Overall Combination: MCE+EC+EP

Finally, we integrated EC, EP, and MCE training techniques to refine the ESS space. We first used the MCE training to increase the discrimination; then, we applied the same two-layer tree structure for EC followed by mixture-based and feature-based EP. The result of using the first stage of EC alone is "GD-intraEnv+interEnv" in Table IX. The two types of two-stage ESS spaces were listed in Table XII as "MCE+EC+EP(M)" and "MCE+EC+EP(F)" for mixture-based and feature-based EP, respectively. Again for mixture-based EP, we used a hierarchical tree structure to clustering Gaussian mixture components. For feature-based EP, we partitioned each super-vector according to different types of coefficient components, namely, 13 static, 13 first and 13 second order time derivatives of AFE features [51]. We also list the average WERs and P-values for the two overall combination techniques in Table XIII. The P-values are estimated based on the two combination methods versus "GD -intraEnv+interEnv" in Table IX.

From the results in Table IX and XII, we find both the two combination techniques provide better performance than "GD -intraEnv+interEnv". From Table XIII, similar observations from Table VI are obtained. The two combination methods provide clear improvements under low SNR conditions. We further used dependent t-Test to estimate P-values for the overall 50 testing conditions. The corresponding P-values are 0.066 and 0.019, respectively, for "MCE+EC+EP(M)" and "MCE+EC+EP(F)" versus "GD-intraEnv+interEnv". Thus, we claim that both the two combination methods are better than

"GD-intraEnv+interEnv". Since the concepts of mixture-based and feature-based EP techniques are different, we tested recognition by using an integration of the two EP techniques. However, the integration did not give further improvement over "MCE+EC+EP(F)" alone. We believe that it is due to the limited adaptation statistics needed for the per-utterance compensation mode.

TABLE XII
AVERAGE WERS (IN %) FROM 0DB TO 20DB

	Set A	Set B	Set C	Overall
MCE+EC+EP(M)	4.62	4.99	5.60	4.96
MCE+EC+EP(F)	4.62	4.94	5.56	4.94

TABLE XIII
WER(%) AND P-VALUE FOR OVERALL COMBINATION

dB	WER	P-value	WER	P-value
	MCE+EC +EP(M)	vs. MCE+EC	MCE+EC +EP(F)	vs. MCE+EC
20	0.44	0.209	0.47	0.303
15	0.76	0.463	0.75	0.332
10	1.76	0.273	1.79	0.402
5	4.99	0.383	4.93	0.084
0	16.83	0.021	16.73	0.019

5) Comparison with Other Approaches

We also compared the ESSEM performance with other robust approaches on the Aurora2 task. In this paper, we reported the MLLR [20] and MAPLR [18, 21] results. Since we tested performance in a per-utterance unsupervised mode, the amount of adaptation data was quite limited. Therefore, simple diagonal affine transformations were adopted for both MLLR and MAPLR. Moreover, both MLLR and MAPLR used optimal numbers of affine transformations with respect to the available adaptation statistics. During testing, we first used the AGI process followed by a speaking environment cluster selection as shown in (17) to locate one set of HMMs. Then, we further adapted the parameters of the selected HMMs to match the testing condition. To have a fair comparison with ESSEM, we did not adapt variance parameters in the HMMs. For MAPLR, we chose a matrix variate normal prior density [18], and the hyperparameters of the prior density were estimated from the multicondition training set. To achieve better performance, each node in the two-layer tree had its own set of hyperparameters. Accordingly, the cluster selection process not only located an HMM set but also chose a set of hyperparameters that were more relevant to the testing condition. Table XIV lists results of both approaches for the three testing sets. Detailed WERs of MLLR and MAPLR and P-values of "MCE+EC+EP(F)" in Table XII versus them are listed in Table XV.

From Table XII and Table XIV, we found that "MCE+EC+EP(F)" achieves better performance than MLLR and MAPLR in Set A, Set B, and Overall conditions, while MLLR and MAPLR give better performance than ESSEM in Set C. As mentioned earlier, this should be a limitation of MCE training and can be enhanced by using a better online mapping function [46]. From Table XIII and Table XV, we can observe that ESSEM achieves better performance under noisier conditions (0dB and 5dB). We also used the dependent t-Test to estimate the P-values for the overall 50 testing conditions. The P-values are 0.003 and 0.005 for "MCE+EC+EP(F)" in Table XII versus MLLR and MAPLR, respectively. The small P-values confirm that "MCE+EC+EP(F)" is better than both MLLR and MAPLR for this task.

TABLE XIV
AVERAGE WERS (IN %) FROM 0DB TO 20DB

	Set A	Set B	Set C	Overall
MLLR	4.88	5.22	5.53	5.14
MAPLR	4.87	5.13	5.54	5.11

TABLE XV
WER(%) AND P-VALUE FOR MLLR AND MAPLR

dB	WER	P-value	WER	P-value
	MLLR	ESSEM vs. MLLR	MAPLR	ESSEM vs. MAPLR
20	0.48	0.480	0.48	0.476
15	0.76	0.329	0.76	0.404
10	1.86	0.029	1.81	0.264
5	5.21	0.002	5.11	0.018
0	17.42	0.013	17.40	0.004

From above experiments, we observed that our optimal offline configuration for ESSEM is an integration of EC, EP, and MCE training. To have a complete comparison, we also tested the ESSEM performance with the optimal configuration using MFCC with CMS [8, 9] and simple back-end topology [13] (the same to the configuration used in Section 5.1). Experimental results are listed in Appendix.

VI. Conclusion

We present an ESSEM framework that can be applied to enhance performance robustness of ASR under noisy conditions. We also propose techniques to refine the ESS spaces for ESSEM and thereby enhance its performance. We first introduce EC and EP to structure the ESS space well; then, we propose intraEnv and interEnv training to improve environment discriminative power. We tested the ESSEM performance with its extensions in an unsupervised compensation (self-learning) mode with very limited adaptation data. For EC, although it requires an online cluster selection process before stochastic matching, the dimensionality of the selected sub-space is smaller than the original space. The computational cost is therefore lower than the original method. Moreover, the selected sub-space can provide higher resolution to model the target super-vector for the testing environment than the entire ESS space. For EP, the parameters belonging to different groups are estimated individually, and therefore the overall estimation of super-vector can be obtained accurately. Although we need to conduct several stochastic matching procedures instead of once, partitioning high-dimensional super-vectors is favorable in applications with limited resources of online operation. Next, we use intraEnv and interEnv training algorithms to enhance confidence interval within one particular environment and increase distance across different environments, respectively. Recognition results indicate that ESSEM achieves better

performance with an MCE-trained ESS space than an ML-trained ESS space on both GI and GD systems. We also adopt two measurements to directly investigate the effects of intraEnv and interEnv training. We show that intraEnv training enhances the separation between parameters within an HMM set for a particular environment, while interEnv training increases the difference across environments. Finally, we integrate all techniques, namely MCE training with EC and EP, to obtain our best environment configuration.

In this study, we implemented the ESSEM framework with an ESS space formed by 34 different environments in the offline phase. We believe the same approach can be extended to more environments for different ASR tasks. Moreover, we focus on the offline preparation issues in this paper; many online super-vector estimation issues are also critical to the ESSEM performance and will be further studied.

APPENDIX

In this section, we present the ESSEM performance using our best offline configuration—an integration of EC, EP, plus intraEnv and interEnv training—with MFCC plus per-utterance CMS [8, 9] as the front-end processing and simple back-end [13] as the HMM model topology. The front-end and back-end configurations are the same to Table I to Table VI in Section 5.1. The online mapping function is the linear combination function as presented in (26), and the results to be compared with are those listed in Table V. By comparing "GD -EC(2)+EP(F)" in Table V and "GD-MCE+EC(2)+ EP(F)" in Table XVI, we can clearly observe that the improvement achieved by the MCE-based intraEnv and interEnv training. The WER reduction for the overall condition is 12.50% (7.84% WER to 6.86% WER).

TABLE XVI
AVERAGE WERS (IN %) FROM 0DB TO 20DB

	Set A	Set B	Set C	Overall
GD-MCE+EC(2)+EP(F)	6.99	6.66	7.00	6.86

ACKNOWLEDGMENT

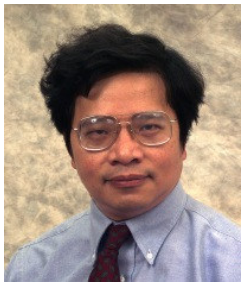
The authors would like to thank Jinyu Li and Chengyuan Ma at the Georgia Institute of Technology for their helpful discussions, and the anonymous reviewers for their valuable feedback.

REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp.257-286, Feb. 1989.
[2] B.-H. Juang, and L. R. Rabiner, "The segmental K-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. Speech Audio Processing*, vol. 38, pp. 1639-1641, 1990.
[3] B.-H. Juang, W. Chou, C.-H. Lee, "Statistical and discriminative methods for speech recognition," In: C.-H. Lee, K.K. Soong, F.K. Paliwal, Automatic Speech and Speaker Recognition: Advanced Topics, Chapter 5. Kluwer Academic Publishers, Dordrecht, 1996.
[4] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp.190-202, May1996.
[5] A. C. Suredran, C.-H. Lee, and M. Rahim, "Nonlinear compensation for stochastic matching," *IEEE Trans. Speech Audio Processing*, vol. 7, pp.643-655, Nov.1999.
[6] D. V. Compernolle, "Noise adaptation in a hidden Markov model speech recognition system," *Comput. Speech and Lang.*, vol. 3, pp.151-167, 1989.
[7] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoustics, Speech, Signal Processing,* vol. 27, pp.113-120, Apr. 1979.
[8] H. Kim and R. C. Rose, "Cepstrum-domain acoustic feature compensation based on decomposition of speech and noise for ASR in noisy environments," *IEEE Trans. Speech Audio Processing*, vol. 11, no. 5, pp. 435-446, 2003.
[9] B. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, pp. 1304-1312, Jun. 1974.
[10] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoustics., Speech, Signal Processing*, vol. 29, pp. 254-272, 1981.
[11] C.-W. Hsu and L.-S. Lee, "Higher order cepstral moment normalization (HOCMN) for robust speech recognition," in *Proc. ICASSP 2004*, pp.197-200, 2004.
[12] Y. H. Suk, S. H. Choi, and H. S. Lee, "Cepstrum third-order normalization method for noisy speech recognition," Electronics Letters, vol. 35, no. 7, pp. 527-528, Apr. 1999.
[13] D. Macho, L. Mauuary, B. Noe, Y. M. Cheng, D. Ealey, D. Jouver, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases," in *Proc. ICSLP 2002*, 2002.
[14] A. Acero, "Acoustical and environmental robustness in automatic speech recognition," *Ph.D. Dissertation*, ECE, Department, CMU, Sept. 1990.
[15] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 11, pp.568-580, Nov. 2003.
[16] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Technical report, TR 291, Cambridge University*, 1997.
[17] K. Visweswariah, V. Goel, and R. Gopinath, "Structuring linear transforms for adaptation using training time information," in *Proc. ICASSP 2002*, pp. 585-588, 2002.
[18] C.-H. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition", *Proc. IEEE,* vol. 88, pp. 1241-1269, Aug. 2000.
[19] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Processing*, vol. 2, pp.291-99, Apr. 1994.
[20] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech and Lang.*, vol. 9, pp.171-185, 1995.
[21] O. Siohan, C. Chesta, and C.-H. Lee, "Hidden Markov model adaptation using maximum *a posteriori* linear regression," in *Proc. Workshop Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, 1999, pp. 147-150.
[22] Y. Gong, "A method of joint compensation of additive and convolutive distortions for speaker-independent speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 13, pp. 975-983, 2005.
[23] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. ICSLP 2002*, pp. 869-872, 2000.
[24] K. Shinoda and C.-H. Lee, "A structural Bayes approach to speaker adaptation," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 3, pp. 276-287, Mar. 2001.
[25] O. Siohan, C. Chesta, and C.-H. Lee, "Joint maximum a posteriori adaptation of transformation and HMM parameters," *IEEE Trans. Speech Audio Processing*, pp.417-428, 2001.
[26] R. P. Lippmann, E. A. Martin, and D. B. Paul, "Multi-style training for robust isolated-word speech recognition," in *Proc. ICASSP 1987,* Dallas, TX, Apr. 1987.
[27] Y. Tsao and C.-H. Lee, "A vector space approach to environment modeling for robust speech recognition," in *Proc. ICSLP 2006*, pp.785-788, Sept. 2006.
[28] K.-T. Chen, W.-W. Liau, H.-M. Wang, and L.-S. Lee, "Fast speaker adaptation using eigenspace-based maximum likelihoods linear regression," in *Proc. ICSLP 2000*, 2000.
[29] A. P. Dempster, N. M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, pp. 1-38, 1977.
[30] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 257-265, 1997.
[31] Y. Tsao and C.-H. Lee, "Two extensions to ensemble speaker and speaking environment modeling for robust automatic speech recognition," in *ASRU*, Dec. 2007.
[32] S. Chen and D. Donoho, "Basis pursuit," in *Proc. Conf. Signals, Syst.*

Comput., pp. 41-44, 1994.
[33] G. Meinardus, G. Nurnberger, M. Sommer, and H. Strauss, "Algorithms for piecewise polynomials and Splines with free knots," in *Math. of Comput.*, vol. 53, pp. 235-247, 1989.
[34] Y. Tsao, S.-M. Lee and L.-S. Lee, "Segmental eigenvoice with delicate eigenspace for improved speaker adaptation," *IEEE Trans. Speech Audio Processing*, vol.13, pp.399-411, 2005.
[35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, New York, Wiley, 2001.
[36] V. Valtchev, J. Odell, P. C. Woodland and S. Young, "MMIE training of large vocabulary recognition systems," *Speech Communication*, vol. 22, no. 4, pp. 303-314, 1997.
[37] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Proc. ICASSP 2002*, pp. I105-I108, 2002.
[38] J. Li, "Soft margin estimation for automatic speech recognition," *Ph.D. Dissertation*, School of ECE, Georgia Institute of Technology, 2008.
[39] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proc. IEEE*, vol. 86, pp. 2345-2373, 1998.
[40] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in Eigenvoice space," *IEEE Trans. Speech Audio Processing*, vol. 8, pp.695-707, Nov. 2000.
[41] T. J. Hazen, "A comparison of novel techniques for rapid speaker adaptation," *Speech Communication*, vol. 31, pp.15-33, May 2000.
[42] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, and P. Laface, "Channel factors compensation in model and feature domain for speaker recognition," in *IEEE Odyssey Workshop*, Jun. 2006.
[43] Y. Tsao and C.-H. Lee, "An ensemble modeling approach to joint characterization of speaker and speaking environments," in *Proc. Interspeech 2007*, Aug. 2007.
[44] B. Mak, T.-C. Lai, and R. Hsiao "Improving reference speaker weighting adaptation by the use of maximum-likelihood reference speakers," in *Proc. ICASSP 2006*, vol. 1, pp. 229-232, May 2006.
[45] D. Pearce and H.-G. Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR'2000*, Paris, France, 2000.
[46] Y. Tsao and C.-H. Lee, "Improving the ensemble speaker and speaking environment modeling approach by enhancing the precision of the online estimation process," in *Proc. Interspeech 2008*, 2008.
[47] M. J. F. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Processing*, vol. 8, no. 4, pp. 417-428, Apr. 2000.
[48] A. J. Hayter, *Probability and Statistics for Engineers and Scientists*, Duxbury Press, 2006.
[49] A. Agresti and C. A. Franklin, *Statistics: The Art and Science of Learning from Data* (*MyStatLab Series*), Prentice Hall, 2008.
[50] Y. Onishi and K.-I. Iso, "Speaker adaptation by hierarchical eigenvoice," in *Proc. ICASSP 2003*, vol. 1, pp. 576-579, Apr. 2003.
[51] J. Wu and Q. Huo, "Several HKU approaches for robust speech recognition and their evaluation on Aurora connected digit recognition tasks," in *Proc. Eurospeech 2003,* 2003.
[52] Y. Tsao, J. Li, and C.-H. Lee, "A study on separation between acoustic models and its applications," in *Proc. Interspeech 2005*, pp. 1109-1112, 2005.

Yu Tsao received his B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, in 1999 and 2001, respectively, and the Ph.D. degree at the school of electrical and computer engineering, Georgia Institute of Technology, Atlanta, in 2008. His research interests focus on detection-based speech recognition, noise robustness, speaker adaptation, and speaker recognition.



Chin-Hui Lee is a professor at School of Electrical and Computer Engineering, Georgia Institute of Technology. Dr. Lee received the B.S. degree in Electrical Engineering from National Taiwan University, Taipei, in 1973, the M.S. degree in Engineering and Applied Science from Yale University, New Haven, in 1977, and the Ph.D. degree in Electrical Engineering with a minor in Statistics from University of Washington, Seattle, in 1981.

Dr. Lee started his professional career at Verbex Corporation, Bedford, MA, and was involved in research on connected word recognition. In 1984, he became affiliated with Digital Sound Corporation, Santa Barbara, where he engaged in research and product development in speech coding, speech synthesis, speech recognition and signal processing for the development of the DSC-2000 Voice Server. Between 1986 and 2001, he was with Bell Laboratories, Murray Hill, New Jersey, where he became a Distinguished Member of Technical Staff and Director of the Dialogue Systems Research Department. His research interests include multimedia communication, multimedia signal and information processing, speech and speaker recognition, speech and language modeling, spoken dialogue processing, adaptive and discriminative learning, biometric authentication, and information retrieval. From August 2001 to August 2002 he was a visiting professor at School of Computing, The National University of Singapore. In September 2002, he joined the Faculty Georgia Institute of Technology.

Prof. Lee has participated actively in professional societies. He is a member of the IEEE Signal Processing Society (SPS), Communication Society, and the International Speech Communication Association (ISCA). In 1991-1995, he was an associate editor for the IEEE Transactions on Signal Processing and Transactions on Speech and Audio Processing. During the same period, he served as a member of the ARPA Spoken Language Coordination Committee. In 1995-1998 he was a member of the Speech Processing Technical Committee and later became the chairman from 1997 to 1998. In 1996, he helped promote the SPS Multimedia Signal Processing Technical Committee in which he is a founding member.

Dr. Lee is a Fellow of the IEEE, and has published more than 250 papers and 25 patents on the subject of automatic speech and speaker recognition. He received the SPS Senior Award in 1994 and the SPS Best Paper Award in 1997 and 1999, respectively. In 1997, he was awarded the prestigious Bell Labs President's Gold Award for his contributions to the Lucent Speech Processing Solutions product. Dr. Lee often gives seminal lectures to a wide international audience. In 2000, he was named one of the six Distinguished Lecturers by the IEEE Signal Processing Society. He was also named one of the two ISCA's inaugural Distinguished Lecturers in 2007-2008. Recently he won the SPS's 2006 Technical Achievement Award for "Exceptional Contributions to the Field of Automatic Speech Recognition".