ISSN No. 0976-5697

Volume 2, No. 6, Nov-Dec 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Speaker Ensembles Recognition In Different Noisy Environments

Ketha. Sriramachandram*[1], K. Ramanjaneyulu [2]
*[1]M tech (DECS), [2]Assoc.Proofessor,
Department of ECE, QISCET, Ongole,
Prakasam (dt), A.P.,India
*sriram.ketha@gmail.com

Abstract— The recognition 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.

Index Terms—Environment modeling, noise robustness.

I. INTRODUCTION

The performance of automatic speech recognition (ASR) systems has improved significantly by adopting the hidden Markov model (HMM) was as a fundamental tool to model speech signals. 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

a. Speaker effects, e.g., speech production, accent, dialect, and speaking rate differences and
b. 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.

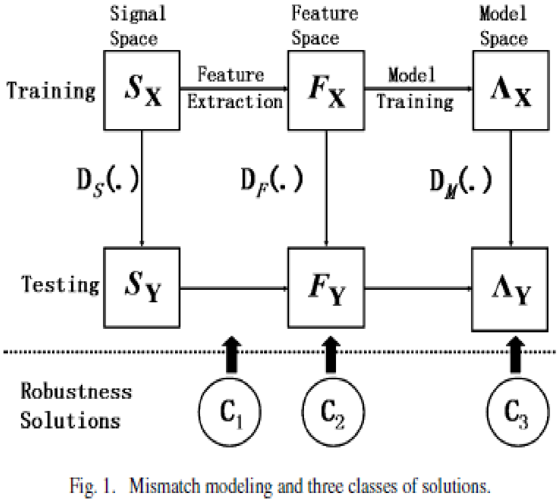


Fig. 1. Mismatch modeling and three classes of solutions.

The mismatch between training and testing conditions can be viewed in the signal, feature or model space, as illustrated in Fig. 1.

First, in the signal space, Sx and Sy denote the speech signals in the training and testing conditions, respectively we represent the distortion observed in the signal space as Ds (.). A following feature extraction procedure converts the speech signals to a few compact and perceptually meaningful features we represent training and testing features as Fx and Fy in Fig. 1. From these features, the statistical models Ax and Ay can then be trained. We denote the distortions observed in the feature and model-spaces as D_F (.) and $_{DM}$ (.) and, respectively.

The approaches that tackle the mismatch problems can be roughly classified into three categories: C1, C_2, and C_3 (Fig. 1).

The first category of C1 approaches is often referred to as speech enhancement These approaches usually involve a new feature extraction procedure. One such C_1 approach, spectral subtraction (SS), and its extensions significantly reduce additive noise by subtracting the noise power spectrum from each speech frame involve a new feature extraction procedure.tracting the noise power spectrum from each speech frame. Likewise, cepstral mean subtraction (CMS), normalizes speech features features in the cepstral domain by subtracting the means from speech frames. Techniques using second or higher order cepstral moment normalization adjusts the distribution of noisy speech features closer to that of the clean ones and provides further improvement over the first-order CMS. More recently, the ETSI advanced front-end (AFE) is proposed to achieve good performance in ASR noise robustness. This ETSI AFE removes mismatch by using several stages of noise reduction schemes, including a two-stage Wiener filter,

Signal-to-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 DF(.) 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 and the stereo-based piecewise linear compensation environments (SPLICE) technique, for example, perform feature compensation with a correction vect or, which is estimated or located with a VQ code word that indicates the gap between the training and testing environments. Similarly, both feature-space maximum-likelihood linear regression (MLLR) and feature space Eigen-MLLR compute affine transformations to compensate noisy features based on a maximum-likelihood (ML) criterion.

The third class of approaches, C3, 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, Ax, to a new set of acoustic models Ay 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. 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 is a well-known method belonging to this category. On the other hand, indirect adaptation models the difference between Ax and Ay training and testing conditions by a mapping function that transforms the original models to new models.

The most often used form of the mapping function is an affine transformation. Maximum-likelihood linear regression (MLLR) and its Bayesian version, maximum *a posteriori* linear regression (MAPLR) [3], have been adopted with good success, where the affine transformations are estimated through ML and MAP learning, respectively. Moreover, stochastic matching [1], [2] 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 and. A vector Taylor series (VTS) expansion is often used to approximate the distortion model. Examples include the joint compensation of additive and convoluted distortion (JAC) and VTS-based HMM adaptation [4].

When comparing the direct and indirect adaptation approaches, the later ones are generally more effective features. 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) [5] uses such a hierarchical structure and shows performance improvements over the conventional MAP. Moreover, a unified framework for a joint MAP adaption of transformation (indirect) and HMM (direct) parameters has been proposed 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, 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.

II. ENSEMBLE SPEAKER AND SPEAKING ENVIRONMENT MODELING FRAMEWORK

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. Similar frameworks to indirect ESSEM show good performance in speaker adaptation In this paper, we limit our discussion on direct ESSEM.The proposed ESSEM approach is derived from the stochastic matching framework [1], [2]. 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 Ax and a set of testing data Fy as in Fig. 1, we want to decode a word sequence such that

$$\boldsymbol{W} = \{W_1, W_2, \ldots, W_L\}$$

$$\boldsymbol{W}' = \underset{\boldsymbol{W}}{\operatorname{argmax}} P(F_{\mathrm{Y}}|\boldsymbol{W}, \Lambda_{\mathrm{X}})P(\boldsymbol{W}). \qquad (1)$$

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

$$\Lambda_{\mathrm{Y}} = \mathbf{G}_{\phi}(\Lambda_{\mathrm{X}}).$$

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

$$(\phi', \boldsymbol{W}') = \operatorname*{argmax}_{(\phi,\boldsymbol{W})} P(F_{\mathrm{Y}}|\phi, \boldsymbol{W}, \Lambda_{\mathrm{X}})P(\boldsymbol{W}). \qquad (3)$$

An iterative procedure can be used to solve . Since our main interest is to compute the parameters , we remove the dependence of for notational simplicity and rewrite as

$$\phi' = \operatorname*{argmax}_{\phi} P(F_{\mathrm{Y}}|\phi, \Lambda_{\mathrm{X}}). \qquad (4)$$

The nuisance parameters in (4) are estimated based on the expectation maximization (EM) algorithm *B.*

B. Ensemble Speaker and Speaking Environment Modeling:

It is 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.

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 A_p,p=1,2,….,p . The entire set of mean parameters within a set of HMMs is then concatenated into a super-vector $V_{p,}$ p=1,2,….,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(D*M) dimension. These P super-vectors form an ensemble speaker and speaking environment space, $\Omega_{\mathrm{V}} = \{\mathrm{V}_1 \quad \mathrm{V}_2 \ldots \mathrm{V}_P\},$ 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 for a testing environment with the ESS space prepared in the offline phase

$$\mathrm{V}_{\mathrm{Y}} = \mathbf{G}_{\phi}(\Omega_{\mathrm{V}}) \qquad (5)$$

$$\phi' = \operatorname*{argmax}_{\phi} P(F_{\mathrm{Y}}|\phi, \Omega_{\mathrm{V}}). \qquad (6)$$

Similar to (4), we can use an EM algorithm to estimate the nuisance parameters φ .

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.

A. Structuring the Environment Spaces:

The objective of environment clustering (EC) resembles that of well-known subset selection methods 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 that approximate complicated functions with local polynomial representations.

a. 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 subspace. 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 subspace. For a hierarchical tree with C^1 groups (including the root node, intermediate nodes, and leaf nodes), we can categorize the original ESS space in (5) into C^1subspaces:

$$\Omega_{\mathrm{V}}\,\{\Omega_{\mathrm{V}(1)} \bigcup \Omega_{\mathrm{V}(2)} \cdots \bigcup \Omega_{\mathrm{V}(C)}\}.$$

We specify a function to determine a representative super-vector for each of these R(.) subspaces; for example, the super-vector, $\mathrm{V}_{\mathbf{rep}}^{(c)},$ represents the th cluster $\Omega_{\mathrm{V}(c)}$.

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

More details about the establishment of a hierarchical tree and the calculation of the function R(.) can be found in our previous study [6]. 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 subspace $\Omega_{\mathrm{V}(p)}$ noise subspace$\Omega_{\mathrm{V}(b)}$ and channel subspace, $\Omega_{\mathrm{V}(h)}$

$$\Omega_{\mathrm{V}} = \{\,\Omega_{\mathrm{V}(p)} \quad \bigcup \Omega_{\mathrm{V}(b)} \quad \bigcup \Omega_{\mathrm{V}(h)}\,\}. \qquad (8)$$

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.

b. 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 subvectors. Then, we collect each set of sub-vectors among all the training environments to form a subspace. From our previous study , 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 [5], 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 to measure the similarity between a pair of Gaussian mixture components. Then, the original super-vector is partitioned into sets of sub-vectors $(\mathrm{v}_P = [\mathrm{v}_{P,1}^T, \mathrm{v}_{P,2}^T, \ldots, \mathrm{v}_{P,S}^T]^T,$ for the P th super-vector). Each set of such sub-vectors from the environments then forms a subspace individually, $\Omega_{\mathrm{V}_s} = [\mathrm{V}_{1,s}^T, \mathrm{V}_{2,s}^T, \ldots, \mathrm{V}_{P,s}^T]^T, s = 1,2,\ldots S.$

Another tying method is to classify models with whole-word, or subword units, and accordingly their Gaussian mixture components, into different clusters based on acoustic or linguistic knowledge.

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 subvectors.$(\mathrm{V}_P = [\mathrm{V}_{P,1}^T, \mathrm{V}_{P,2}^T, \ldots, \mathrm{V}_{P,Z}^T]^T,$ for the Pth super-vector). Then, we can construct Z sets of subspaces $\Omega_{\mathrm{V}_z} = \{\mathrm{V}_{1,z}\mathrm{V}_{2,z}, \ldots, \mathrm{V}_{P,z}\}, z = 1,2,\ldots,Z,$ v with each subspace spanned by a particular group of coefficients.

B. *Increasing Coverage of the Environment Spaces:*

Traditionally, discriminative training methods, such as minimum classification error (MCE) , maximum mutual information estimation (MMIE) [7], minimum word/phone error (MWE/MPE) and soft margin estimation (SME) [8], 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 [6] 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.

a. *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 Pth environment, we have the objective function

$$L(\mathrm{V}_P) = \frac{1}{U_P}\sum_{u=1}^{U_P}\frac{1}{1+exp(-\gamma d\left(F_P^u, \mathrm{V}_P, \Psi\right)+\theta)} \qquad (9)$$

where is the F_P^u th training utterance for the th environment; both and 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

$$d\left(F_P^u, \mathrm{V}_P, \Psi\right) = -\bar{g}\left(F_P^u, \mathrm{V}_P, \Psi, W_c\right) + \bar{G}\left(F_P^u, \mathrm{V}_P, \Psi\right) \qquad (10)$$

$$\bar{G}\left(F_P^u, \mathrm{V}_P, \Psi\right) = \frac{1}{\eta}\log\left\{\frac{1}{N}\sum_{n=1}^{N}\exp\left[\eta \times \bar{g}\left(F_P^u, \mathrm{V}_P, \Psi, W_n\right)\right]\right\} \qquad (11)$$

where is a positive control parameter, is the given correct where η is a positive control parameter,W_c is the given correct transcription, and $\{W_1, \ldots, W_N\}$ are the N -best decoded competing word sequences. The -best are generated by decodingF_P^u using the HMMs for the P^th environment. We used a log-likelihood for the discrimination function,$\bar{g}(\cdot)$ in and (11), and adopted the generalized probabilistic descent (GPD) algorithm to update parameters inV_p iteratively

$$\mathrm{V}_P(t+1) = \mathrm{V}_P - \kappa\nabla L(V_P)\,_{\mathrm{V}_P=\mathrm{V}_P(t)} \qquad (12)$$

where κ is a step size.

b. *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 of a total of U utterances for P different environments and define an objective function

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

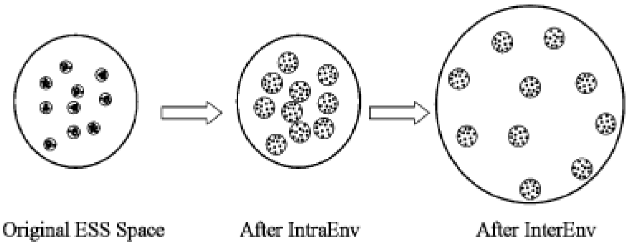


Fig. 2. IntraEnv and interEnv training to increase environment space discrimination.

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

$$d\left(F_{\mathrm{train}}^u, \Omega_{\mathrm{V}}, \Psi\right) = -\bar{g}\left(F_{\mathrm{train}}^u, \Omega_{\mathrm{V}}, \Psi, W_c\right) + \bar{G}\left(F_{\mathrm{train}}^u, \Omega_{\mathrm{V}}, \Psi\right) \qquad (14)$$

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

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 nth most competitive environment to that target environment. Parameters in the ESS space are then updated iteratively based on

W_c

$$\Omega_V(t+1) = \Omega_V(t) - \kappa \nabla L(\Omega_V)_{\Omega_v = \Omega_V(t)}. \qquad (16)$$

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.

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, whose representative super-vector has the highest likelihood to the testing data F_y

$$\Omega_{V(c)} = \arg\max_{c} P(F_Y | R(\Omega_{V(c)})). \qquad (17)$$

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

$$V_Y = G_\phi(\Omega_{V(c)}). \qquad (18)$$

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 subspace $\Omega_{V(h)}$ and estimate the target super-vector through $V_Y = G_\phi(\Omega_{V(h)})$. Previous studies have verified such distortion-specific compensation operating well in speaker variations channel mismatches and additive background noise. It is also useful if each distortion has individually refined structure, such as hierarchical structure to facilitate estimating the target super-vector in

B. *Environment Partitioning:*

For the mixture-bases EP technique, we estimate sub-vectors in Vy through stochastic matching as shown in individually.

For example, the th sub-vector is estimated by

$$V_{Y,s} = G_{\phi_s}(\Omega_{V_s}), \quad s = 1, 2, \ldots, S. \qquad (19)$$

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

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

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 th group of coefficients, we have

$$V_{Y,z} = G_{\phi_z}(\Omega_{V_z}), \quad z = 1, 2, \ldots, Z. \qquad (21)$$

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

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

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: 5, 10, 15, and 20 dB, 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-adaptation (unsupervised) mode is used, and each testing utterance was first decoded into an -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 [10]; in this paper, we selected the linear combination function throughout the following experiments. This mapping function is also used in cluster adaptive training (CAT) [11] and Eigen voice .With the EC algorithm, the online super-vector estimation in becomes

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

where $\hat{w}_p$ is the pth weighting coefficient in the linear combination function, and $P^{(c)}$ is the number of bases in the Cth subspace. The set of weighting coefficients is estimated according to the ML algorithm

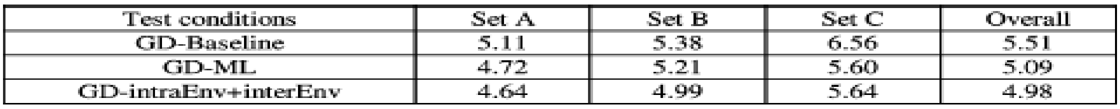
$$\{\hat{w}_p\}_{p=1}^{P^{(c)}} = \arg \max_{\{w_p\}_{p=1}^{P^{(c)}}} P\left(F_Y | \sum_{p=1}^{P^{(c)}} w_p V_p\right). \qquad (24)$$

As mentioned earlier, the online process of EC resembles the subset selection methods. When comparing to CAT and Eigen voice, 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 Pth cluster) to estimate the target super-vector in (5). 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 The representative HMMset can provide more accurate statistics estimation than the environment-independent HMM set.

A. *Environment Clustering and Environment Partitioning:*

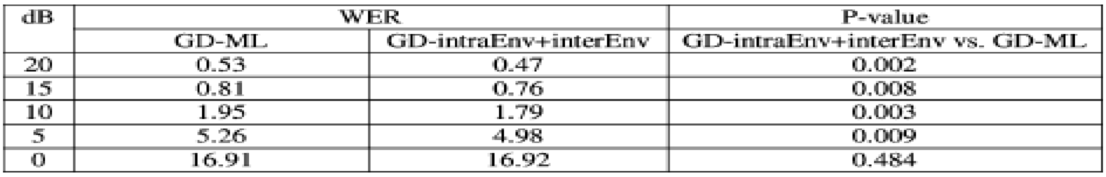
We first evaluated the performance of the EC and EP algorithms. For this set of experiments, each speech frame was characterized by 39 coefficients consisted of 13 MFCCs with their first and second order time derivatives. An utterance-level CMS was performed for normalization. All digits were modeled 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.

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



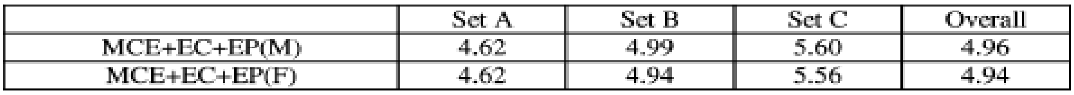
Test conditions	Set A	Set B	Set C	Overall
GD-Baseline	5.11	5.38	6.56	5.51
GD-ML	4.72	5.21	5.60	5.09
GD-intraEnv+interEnv	4.64	4.99	5.64	4.98

TABLE X
WER(%) AND P-VALUE FOR ESSEM WITH ML AND MCE TRAINED ESS SPACES



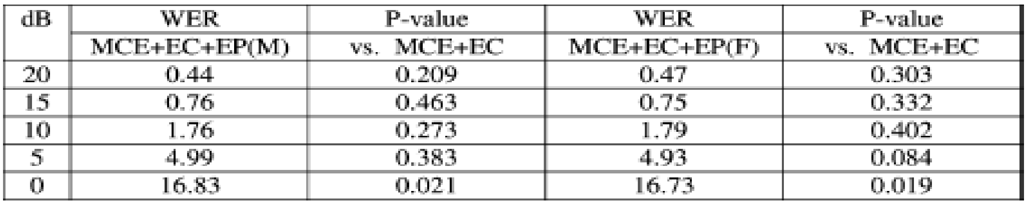
dB	WER		P-value
	GD-ML	GD-intraEnv+interEnv	GD-intraEnv+interEnv vs. GD-ML
20	0.53	0.47	0.002
15	0.81	0.76	0.008
10	1.95	1.79	0.003
5	5.26	4.98	0.009
0	16.91	16.92	0.484

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



	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

We tested ESSEM on gender-independent (GI) and gender-dependent (GD) systems. For the GI system, a GIHMM 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 GDHMMsets were first trained. Then, 17 environment-specific HMMsets 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.

a. 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 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 .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 intraEnv” in Table IX.

From the results in Tables IX and XII, we find both the two combination techniques provide better performance than “GD-intraEnv intraEnv.” From Table XIII,. 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 intraEnv.” Thus, we claim that both the two combination methods are better than “GD-intraEnv intraEnv.” 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.

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 subspace is smaller than the original space. The computational cost is therefore lower than the original method. Moreover, the selected subspace 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, 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 HMMset 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 paper, 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 supervector estimation issues are also critical to the ESSEM performance and will be further studied.

VII. REFERENCES

[1]. A. Sankar and C.-H. Lee, “A maximum-likelihood approach to stochastic matching for robust speech recognition,” *IEEE Trans. Speech Audio Process.*, vol. 4, no. 3, pp. 190–202, May 1996.

[2]. A. C. Suredran, C.-H. Lee, and M. Rahim, “Nonlinear compensation for stochastic matching,” *IEEE Trans. Speech Audio Process.*, vol. 7, no. 6, pp. 643–655, Nov. 1999.

[3]. 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.

[4]. A. Acero, L. Deng, T. Kristjansson, and J. Zhang, “HMM adaptation using vector Taylor series for noisy speech recognition,” in *Proc. ICSLP’02*, 2000, pp. 869–872.

[5]. K. Shinoda and C.-H. Lee, “A structural Bayes approach to speaker adaptation,” *IEEE Trans. Speech Audio Process.*, vol. 9, no. 3, pp. 276–287, Mar. 2001.

[6]. Y. Tsao and C.-H. Lee, “Two extensions to ensemble speaker and speaking environment modeling for robust automatic speech recognition,” in *Proc. ASRU*, Dec. 2007, pp. 77–80.

[7]. V. Valtchev, J. Odell, P. C. Woodland, and S. Young, “MMIE training of large vocabulary recognition systems,” *Speech Commun.*, vol. 22,no. 4, pp. 303–314, 1997.

[8]. J. Li, “Soft margin estimation for automatic speech recognition,” Ph.D. dissertation, School Elect. Comput. Eng., Georgia Inst. Technol., Atlanta, GA, 2008.

[9]. Y. Tsao and C.-H. Lee, “An ensemble modeling approach to joint characterization of speaker and speaking environments,” in *Proc. Interspeech 2007*, Aug. 2007, pp. 1050–1053.

[10]. 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’08*, 2008, pp. 1265–1268.

[11]. M. J. F. Gales, “Cluster adaptive training of hidden Markov models,”*IEEE Trans. Speech Audio Process.*, vol. 8, no. , pp. 417–428, Jul. 2000.