Improving the Ensemble Speaker and Speaking Environment Modeling Approach by Enhancing the Precision of the Online Estimation Process

Yu Tsao and Chin-Hui Lee

School of Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, GA 30332-0250, USA
{yutsao, chl}@ece.gatech.edu

Abstract

In this paper, we study methods to enhance the precision of the online estimation process of a recently proposed approach, ensemble speaker and speaking environment modeling (ESSEM), and therefore improve its overall performance. The ESSEM approach consists of two integral phases, offline and online. In the offline phase, an ensemble environment configuration is prepared by a large collection of acoustic models. Each set of acoustic models represents a particular environment. In the online phase, with speech data from the testing condition, we estimate a mapping function and use it to generate a new set of acoustic models for that particular testing condition. In our previous study, we have discussed the issues of the offline process and proposed algorithms to refine the environment configuration. In this paper, we first study different online mapping structures and compare their performances on a same environment configuration. Next, we propose a multiple clustering matching algorithm to further improve the overall performance of ESSEM. We tested ESSEM and its extensions on the full evaluation set of the Aurora2 connected digit recognition task. When using our best offline environment configuration along with a properly specified online estimation method, the ESSEM approach can achieve an average word error rate (WER) of 4.77%, corresponding to a WER reduction of 13.43% (from 5.51% WER to 4.77% WER) over the baseline result.

Index Terms: acoustic modeling, robust speech recognition

1. Introduction

For automatic speech recognition (ASR), robust performance under unknown testing conditions is a key issue for its success. In a real-world ASR application, a testing condition usually contains multiple distortion sources that may come from: 1) speaker effects–age, gender, and accent differences; 2) speaking environment effects–interfering noise, channel, and transducer distortions. Although some functions can characterize particular distortions well, the form of an unknown combination of speaker and speaking environment distortion sources is generally complex and hard to specify.

Many approaches have been proposed to enhance the ASR robustness. Among them, a category of approaches maps the original acoustic models to a new set of acoustic models for the testing condition. The model-mapping process can be done either directly or indirectly [1]. Maximum a posteriori (MAP) estimation [2] is a well-known direct mapping approach that adjusts the parameters of the acoustic models directly. On the other hand, the indirect mapping approaches, such as–maximum likelihood linear regression (MLLR) [3] and its Bayesian version, maximum a posteriori linear regression (MAPLR) [4], and stochastic matching [5] [6]–adopt a mapping function to transform the parameters of the acoustic models to match the testing features.

Recently, an ensemble speaker and speaking environment modeling (ESSEM) approach has been proposed [7] [8]. The ESSEM approach is derived from the stochastic matching algorithm [5] [6]. However, instead of using one set of acoustic models in stochastic matching, ESSEM prepares a large collection of acoustic models to effectively represent the complex structure of the environment space. The ESSEM framework comprises two stages, the offline and online phases. In the offline phase, ESSEM prepares an ensemble speaker and speaking environment configuration; in the online phase, ESSEM estimates a mapping function and uses it to obtain the acoustic models for the testing condition. In our previous study, we have proposed algorithms to refine the environment configuration in the offline phase [8]. In this paper, we focus on the online issues and present methods to enhance the precision of the online environment modeling.

We conducted our experiments on the Aurora2 database [9]. We used the environment configuration that achieves the best performance reported in [8] and compared the overall performances of ESSEM using different online methods on a gender dependent (GD) system [8]. With our best offline and online settings, ESSEM achieves an average of 4.77% word error rate (WER) on the full evaluation set of the Aurora2 task.

2. Review of the ESSEM Approach

In this section, we first review the ESSEM approach. Then, we introduce the previously proposed environment clustering (EC) algorithm to refine the environment configuration [8].

2.1. The ESSEM Framework

In this sub-section, we detail the two phases of the ESSEM framework, namely the offline environment configuration preparation and the online super-vector estimation.

2.1.1. Offline Environment Configuration Preparation

In the offline phase of the ESSEM framework, we collect speech data from a wide range of different speaker and speaking environments. Since the collection of real-world data can be prohibitive, we may use the Monte Carlo (MC) methods [10] to artificially simulate the training sets [7]. By using the MC methods, we can further control the constitution and coverage of the constructed environment configuration. After we have collected P sets of training data, we can train P sets of hidden Markov models (HMMs), Λ_p, p=1,…, P, for P different environments. For ease of modeling, the entire set of mean parameters for each Gaussian within a set of HMMs is concatenated into a super-vector, V_p, p=1,…, P. These P super-vectors form an ensemble speaker and speaking (ESS) environment space, $\Omega_V=\{V_1\ V_2 \ldots V_P\}$. We call this

environment space the ESS space for notational simplicity.

2.1.2. Online Super-vector Estimation

In the online phase, ESSEM estimates the target super-vector, V_Y, for the testing environment with the ESS space through a mapping function, $\mathbf{G}_\varphi$:

$$V_Y = \mathbf{G}_\varphi(\Omega_V), \quad (1)$$

with

$$\varphi' = \underset{\varphi}{argmax}\ P(F_Y \mid \varphi, \Omega_V), \quad (2)$$

where F_Y is the speech data from the testing condition, and φ represents the nuisance parameters in the mapping function. The nuisance parameters are only used in the mapping procedure but not involved in the recognition procedure. We can estimate the nuisance parameters based on the expectation-maximization (EM) algorithm [11]. With the estimated target super-vector, V_Y, we can accordingly obtain the set of acoustic models, Λ_Y, for the testing condition [8].

2.2. Environment Clustering (EC)

The basic concept of environment clustering (EC) resembles that of the well-known subset selection methods [12] [13] that determine a subset of components from the entire set of components to model a signal of interest.

2.2.1. Offline Environment Configuration Preparation

The EC algorithm clusters 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 our previous study [8], we present a hierarchical clustering procedure to construct a tree structure for environment clustering. When the hierarchical tree structure has C nodes (including the root node, intermediate nodes, and leaf nodes), we can categorize the original ESS space in Eq.(1) into C environment clustered sub-spaces: $\Omega_{\mathbf{V}} = \{\Omega_{\mathbf{V}^{(1)}} \cup \Omega_{\mathbf{V}^{(2)}} \ldots \cup \Omega_{\mathbf{V}^{(C)}}\}$.

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

$$V_{rep}^{(c)} = R(\Omega_{\mathbf{V}^{(c)}}). \quad (3)$$

2.2.2. Online Super-vector Estimation

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

$$\Omega_{\mathbf{V}^{(c)}} = arg\ \underset{\tilde{c}}{max}\, P(F_Y \mid R(\Omega_{\mathbf{V}^{(\tilde{c})}})). \quad (4)$$

With the selected cluster, $\Omega_{\mathbf{V}^{(c)}}$, and based on Eq.(1), we estimate the target super-vector, V_Y, through:

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

The nuisance parameters, φ, are estimated based on the stochastic matching algorithm as shown in Eq.(2).

3. Enhancing on Online Estimation

In this section, we study methods to enhance the precision of the online super-vector estimation. We first introduce different parametric functions as the online mapping structures to estimate the target super-vector. Moreover, we propose a multiple cluster matching (MCM) algorithm to improve the EC algorithm in the online phase.

3.1. Online Mapping Structure

Intuitively, by using a more complex mapping function in Eq.(1), ESSEM can better characterize a testing environment. However, too many free parameters to estimate in a complex function may cause an over-fitting problem. Therefore, the best form of mapping structure for a particular task should rely on the amount of available adaptation data. In this paper, we restrict our attention on simple linear mapping functions. More complex mapping functions can be studied in the future.

3.1.1. Best First

The simplest form of mapping function is the best first method. The best first method determines V_Y by locating the most matched super-vector in the ESS space:

$$V_Y = arg\ \underset{p}{max}\, P(F_Y \mid V_p),\ p=1, 2 \ldots P, \quad (6)$$

where P is the number of super-vector bases in the ESS space.

3.1.2. Linear Combination

When using the linear combination function as the mapping structure, Eq.(1) can be represented as:

$$V_Y = \sum_{p=1}^{P} \hat{w}_p V_p, \quad (7)$$

where $\hat{w}_p$ is the p-th weighting coefficient in the linear combination function. We estimate the set of weighting coefficients based on a maximum likelihood (ML) criterion:

$$\{\hat{w}_p\}_{p=1}^{P} = arg\ \underset{\{w_p\}_{p=1}^{P}}{max}\, P(F_Y \mid \sum_{p=1}^{P} w_p V_p). \quad (8)$$

3.1.3. Linear Combination with a Correction Bias

Next, we improve the linear combination function by incorporating a global correction bias $\hat{b}$ into Eq.(7):

$$V_Y = \sum_{p=1}^{P^{(c)}} \hat{w}_p V_p + \hat{b}. \quad (9)$$

Again, the set of weighting coefficients and the correction bias can be estimated base on the ML criterion:

$$\{\{\hat{w}_p\}_{p=1}^{P}; \hat{b}\} = arg\ \underset{\{\{w_p\}_{p=1}^{P}; b\}}{max}\, P(F_Y \mid \sum_{p=1}^{P} w_p V_p + b). \quad (10)$$

3.2. Multiple Cluster Matching

In this sub-section, we present a multiple cluster matching (MCM) algorithm to reduce the performance degradations caused by a possible poor cluster selection process; thereby, enhance the precision of the online super-vector estimation. The basic concept of MCM is similar to that of the ensemble estimator (EE) algorithm [14]. The EE algorithm is developed in the research for sparse representations of signals and usually compared with the subset selection methods [12] [13]. Instead of finding a single best subset, the EE algorithm models the target signal with a combination of estimates obtained from multiple subsets. In particular tasks, the subset selection methods generate unstable results [15], and the EE algorithm can provide better performance in such conditions.

We apply the MCM algorithm into the ESSEM framework in the online phase. When we have prepared an EC-structured

ESS space in the offline phase, instead of performing a CS procedure to determine the most relevant cluster of environments, we estimate a super-vector for each cluster:

$$V_{\mathbf{Y}^{(c)}} = \mathbf{G}_{\varphi}(\Omega_{\mathbf{V}^{(c)}}),\ c=1,2\ldots,C. \quad (11)$$

Then, the collection of all the estimated C super-vectors forms a new ensemble environment space, $\Omega_{\mathbf{V}_E}$:

$$\Omega_{\mathbf{V}_E} = \{V_{\mathbf{Y}^{(1)}} V_{\mathbf{Y}^{(2)}} \ldots V_{\mathbf{Y}^{(C)}}\}. \quad (12)$$

Finally, a stochastic matching process is carried out to estimate the super-vector for the testing condition through a multiple cluster matching (MCM) function, $\mathbf{G}_{\varphi_E}$:

$$V_{\mathbf{Y}} = \mathbf{G}_{\varphi_E}(\Omega_{\mathbf{V}_E}), \quad (13)$$

with

$$\varphi_E' = \underset{\varphi_E}{argmax}\ P(F_{\mathbf{Y}} \mid \varphi_E, \Omega_{\mathbf{V}_E}), \quad (14)$$

where φ_E stands for the set of nuisance parameters in the MCM function. Similarly, the mapping structure of the MCM function, $\mathbf{G}_{\varphi_E}$, can be either the best first method in Eq.(6), linear combination function in Eq.(7), or linear combination with a correction bias function in Eq.(9).

When the number of the training environments grows large, or when the tree structure built by the EC algorithm is complex, we need special strategies to enhance the efficiency of the MCM algorithm. One possible method is to only take account of a subset of clusters in the tree structure. Environments in those clusters have closer acoustic properties to the testing condition. The subset of clusters can be collected by finding those clusters with their representative super-vectors in Eq.(3) giving higher likelihood scores to the testing data. Then, we have a new environment space, $\Omega_{\mathbf{V}_E'}$:

$$\Omega_{\mathbf{V}_E'} = \{V_{\mathbf{Y}^{(1)}} V_{\mathbf{Y}^{(2)}} \ldots V_{\mathbf{Y}^{(C')}}\}, \quad (15)$$

where C' is smaller than C, and $\Omega_{\mathbf{V}_E'}$ is a sub-space of $\Omega_{\mathbf{V}_E}$. Finally, we can obtain the target super-vector, $V_{\mathbf{Y}}$, through:

$$V_{\mathbf{Y}} = \mathbf{G}_{\varphi_E}(\Omega_{\mathbf{V}_E'}). \quad (16)$$

4. Experiments

In this section, we first introduce the experimental framework; then, we present the recognition performance achieved by the ESSEM approach with different online methods.

4.1. Experimental Setup

We evaluated the ESSEM approach on the Aurora2 database [9]. The multicondition training set was used both to train HMMs and to build the ESS spaces. The training set includes 17 different speaking environments that are originated from the same four types of noise as in test set A, at four different SNR levels: 5dB, 10dB, 15dB, 20dB, along with the clean condition. We further divided the training set into two gender-specific subsets. Therefore, we obtained 34 (17×2) speaker and speaking environments. We used the word error rate (WER) to evaluate recognition performance on the full evaluation set that consists of 70 different testing conditions with 1001 utterances in each condition.

We used a modified ETSI advanced front-end (AFE) [16] as suggested in [17] for feature extraction. The log-energy component of each frame was replaced with the C0 coefficient. Every feature vector comprises 13 static components plus their first and second order time derivatives. We followed a complex back-end topology as presented in [16] to train HMMs. All digits were modeled by 16-state whole word models with each state characterized by 20 Gaussian mixture components. The silence and the short pause were modeled by 3 states and 1 state, respectively, with each state characterized by 36 Gaussian mixture components.

We tested ESSEM in a per-utterance unsupervised self-adaptation mode [8] on a gender dependent (GD) system. Each testing utterance was first decoded into an N-best list (N=8) and then used for ESSEM adaptation. For the GD system, two sets of gender-specific HMMs were first trained. Then, 17 sets of environment-specific HMMs for each gender were obtained by adapting mean vectors from that gender-specific HMM set to particular environments. Accordingly, two ESS spaces corresponding to the two gender-specific HMM sets were prepared. The same pair of gender-specific HMM sets were used for an automatic gender identification (AGI) process to determine every speaker's gender.

During performance testing, we used every incoming testing utterance to: 1) identify speaker's gender and select the corresponding gender-specific HMMs; 2) select a more suitable EC-clustered ESS space through the CS process; 3) perform ESSEM in an unsupervised self-adaptation manner; 4) test recognition with the ESSEM-adapted acoustic models.

4.2. Result Analysis

First, we present the baseline results in Table 1. We used the testing result of ESSEM with the EC algorithm to represent the baseline performance of ESSEM. For the EC algorithm, we used a two-layered tree structure to cluster environments by following the clustering procedure reported in [8]. In the first layer, the 34 environments were exactly divided into two groups, each corresponding to one of the two genders. In the second layer, another two groups of environments were classified roughly according to high/low SNR levels.

Since the gender identity was determined by the AGI unit beforehand, the EC algorithm did not need an online CS process in Eq.(4) for the first layer. To have a fair comparison, we used the AGI process followed by a one-layer speaking environment CS process as presented in Eq.(4) to locate one set of HMMs. Then, we directly used the located HMM set, without performing stochastic matching, to test recognition for the "Baseline" result in Table 1. Next, we list the result of ESSEM with the EC algorithm as the "Baseline-ESSEM" result in Table 1. For "Baseline-ESSEM", we first adopted the minimum classification error (MCE) training [18] [8] to increase the discriminative power of the ESS space. Then, we applied the EC algorithm with the two-layer tree to structure the ESS space well. Finally, a linear combination function as shown in Eq.(7) was used as the online mapping structure.

Table 1. Average word error rates (in %) from 0dB to 20dB.

	Set A	Set B	Set C	Overall
Baseline	5.11	5.38	6.56	5.51
Baseline-ESSEM	4.64	4.99	5.64	4.98

4.2.1. Structure of Online Mapping Function

Next, we fixed the same offline ESS space as that used in "Baseline-ESSEM" in Table 1 and evaluated the overall ESSEM performances by using two different online mapping structures–the best first method in Eq.(6) and the linear combination with a correction bias function in Eq.(9). The two results are listed as "Best First" and "LC+bias" in Table 2, respectively. By comparing "Baseline-ESSEM" in Table 1, and "Best First" and "LC+bias" in Table 2, we first note that the best first method gives worse performance than the other

two mapping structures. We believe that it is due to the natural limitation of the best first method–the closest super-vector V_p may still be far from the real V_Y especially when the testing condition is very different from any vector in the collection of super-vectors. Second, we can see that "LC+ bias" achieves the best performance among the three mapping structures. Therefore, we confirm that using a more properly specified online mapping function, ESSEM can achieve better performance. Moreover, by comparing the results of "Baseline-ESSEM" in Table 1 and "LC+bias" in Table 2, we can observe that the major improvements come from test set C, where a channel distortion is added as another acoustic difference. The performance improvement suggests that the additional nuisance parameters (the correction bias) enable ESSEM to more accurately characterize the testing conditions that contain distortions not included in the training set.

Table 2. Average word error rates (in %) from 0dB to 20dB.

	Set A	Set B	Set C	Overall
Best First	4.98	5.22	6.38	5.35
LC+bias	4.62	4.95	5.13	4.85

4.2.2. Structure of Multiple Cluster Matching

Finally, we present the results of the MCM algorithm. We used the same offline ESS space as in Table 1 and Table 2 and fixed the linear combination with a correction bias in Eq.(9) as required online mapping structure. Since the total number of nodes in the two-layer tree was not too large (C=7), we used all the clusters to perform the MCM algorithm. We tested the MCM algorithm with two MCM functions–best first in Eq.(6) and linear combination with a correction bias in Eq.(9). The corresponding results are listed as "LC+bias–BF" and "LC+bias–LC+bias" in Table 3. When comparing "LC+ bias" in Table 2 with "LC+bias–BF" and "LC+bias–LC+bias" in Table 3, we confirm that the MCM algorithm further reduces the ESSEM WER from 4.85% to 4.78% and 4.77%, respectively. We also find that "LC+bias–LC+bias" provides slightly better performance than "LC+bias–BF" in Table 3. Therefore, it is verified that using a more suitable MCM mapping function, ESSEM can produce better overall performance. Finally, by comparing all the experimental results presented above, it is clear that "LC+bias–LC+bias" in Table 3 stands for our best performance of the ESSEM approach, which gives a 13.43% WER reduction (from 5.51% WER to 4.77% WER) over the "Baseline" result in Table 1.

Table 3. Average word error rates (in %) from 0dB to 20dB.

	Set A	Set B	Set C	Overall
LC+bias–BF	4.50	4.95	5.02	4.78
LC+bias–LC+bias	4.48	4.95	5.00	4.77

5. Conclusion

In this paper, we study methods to enhance the precision of the online estimation process of the ESSEM approach. We first present different mapping functions and compare their performances. We observe that by using a more properly specified mapping structure, testing environments can be better characterized. When using our best offline environment setting, along with a linear combination with a correction bias as the mapping function, ESSEM achieves the performance of 4.85% WER on the full testing set of the Aurora2 database. Moreover, we propose a multiple cluster matching (MCM) algorithm to further improve the online modeling process. From the experimental results, we observe that the MCM algorithm enables ESSEM to not only achieve a significant performance improvement of 13.43% WER reduction (from 5.51% WER to 4.77% WER) over the baseline result but also produce a further improvement over the ESSEM without using the MCM algorithm (from 4.85% WER to 4.77% WER).

6. Acknowledgements

This work was supported by a Texas Instruments Leadership University (TILU) grant. We also thank Jinyu Li of Georgia Tech for helpful discussions and comments.

7. References

[1] Lee, C.-H. and Huo, Q., "On adaptive decision rules and decision parameter adaptation for automatic speech recognition", Proc. IEEE, Vol. 88, pp. 1241-1269, 2000.

[2] Gauvain, J.-L. and Lee, C.-H., "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", IEEE Trans. Speech Audio Processing, Vol. 2, no. 2 , pp.291-99, Apr. 1994.

[3] Leggetter, C. and Woodland, P., "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models", Compt. Speech and Lang., pp.171-185, 1995.

[4] Siohan, O., Chesta, C., and Lee, C.-H., "Hidden Markov model adaptation using maximum a posteriori linear regression", in Proc.Workshop Robust Methods for Speech Recognition in Adverse Conditions, Tampere, Finland, pp. 147–150, 1999.

[5] Sankar, A. and Lee, C.-H., "A maximum-likelihood approach to stochastic matching for robust speech recognition", IEEE Trans. Speech Audio Processing, Vol. 4, pp.190-202, May 1996.

[6] Suredran, A. C., Lee, C.-H., and Rahim, M., "Nonlinear compensation for stochastic matching", IEEE Trans. Speech Audio Processing, Vol. 7, pp.643-655, Nov. 1999.

[7] Tsao, Y. and Lee, C.-H., "A vector space approach to environment modeling for robust speech recognition", in Proceedings of ICSLP, pp.785-788, Sept. 2006.

[8] Tsao, Y. and Lee, C.-H., "Two extensions to ensemble speaker and speaking environment modeling for robust automatic speech recognition", in ASRU, Dec. 2007.

[9] Pearce, D. and Hirsch, H.-G., "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions", in Proc. ISCA ITRW ASR'2000, Paris, France, 2000.

[10] Metropolis, N. and Ulam, S., "The Monte Carlo method", J. Amer. Statist Assoc., Vol. 44, pp.335-341, Sept. 1949.

[11] Dempster, A. P., Laird, N. M., and Rubin, D.B., "Maximum likelihood from incomplete data via the EM algorithm", J. Roy. Statist. Soc. B, pp. 1-38, 1977.

[12] Chen, S., Donoho, D., and Saunders, M. A., "Atomic Decomposition by Basis Pursuit", in SIAM J. on Scientific Computing, Vol. 20, No. 1, pp. 33-61, 1998.

[13] Mallat, S. and Zhang, Z., "Matching Pursuit with Time-Frequency Dictionaries", in IEEE Trans. on Signal Processing, Vol. 41, pp. 3397-3415, Dec. 1993.

[14] Bruce, A., Gao, H. Y., and Stuetzle, W., "Wavelet denoising: a comparison of subset-selection and ensemble methods", in Statistica Sinica, Vol. 9, pp. 167-182, 1999.

[15] Breiman, L., "Heuristics of instability and stabilization in model selection", in Annals of Statistics, pp.2350-2383, 1996.

[16] Macho, D., Mauuary, L., Noe, B., Cheng, Y. M., Ealey, D., Jouver, D., Kelleher, H., Pearce, D., and Saadoun, F., "Evaluation of a noise-robust DSR front-end on Aurora databases", in Proc. ICSLP'2002, pp. 17–20, Denver, 2002.

[17] Wu, J. and Huo, Q., "Several HKU approaches for robust speech recognition and their evaluation on Aurora connected digit recognition tasks", in Proc. Eurospeech03, 2003.

[18] Juang, B.-H., Chou, W., and Lee, C.-H., "Minimum Classification Error Rate Methods for Speech Recognition", IEEE Trans. Speech Audio Processing, pp. 257-265, 1997.