Acoust. Sci. & Tech. **29**, 1 (2008) ©2008 The Acoustical Society of Japan

PAPER

Histogram equalization for noise-robust speech recognition using discrete-mixture HMMs

Tetsuo Kosaka*, Masaharu Katoh† and Masaki Kohda‡

Graduate School of Science and Engineering, Yamagata University, 4–3–16, Jonan, Yonezawa, 992–8510 Japan

(*Received 28 February 2007, Accepted for publication 18 September 2007*)

Abstract: In this paper, we introduce a new method of robust speech recognition under noisy conditions based on discrete-mixture hidden Markov models (DMHMMs). DMHMMs were originally proposed to reduce calculation costs in the decoding process. Recently, we have applied DMHMMs to noisy speech recognition, and found that they were effective for modeling noisy speech. Towards the further improvement of noise-robust speech recognition, we propose a novel normalization method for DMHMMs based on histogram equalization (HEQ). The HEQ method can compensate the nonlinear effects of additive noise. It is generally used for the feature space normalization of continuous-mixture HMM (CMHMM) systems. In this paper, we propose both model space and feature space normalization of DMHMMs by using HEQ. In the model space normalization, codebooks of DMHMMs are modified by the transform function derived from the HEQ method. The proposed method was compared using both conventional CMHMMs and DMHMMs. The results showed that the model space normalization of DMHMMs by multiple transform functions was effective for noise-robust speech recognition.

Keywords: Speech recognition, Noise robustness, Hidden Markov model, Discrete HMMs, Histogram equalization

PACS number: 43.72.Ne [doi:10.1250/ast.29.66]

1. INTRODUCTION

In recent speech recognition systems, continuous-mixture hidden Markov models (CMHMMs) have been used as acoustic models. The parameters of CMHMMs can be estimated efficiently under the assumption of a normal distribution. Meanwhile, discrete HMMs (DHMMs) based on vector quantization (VQ) have a problem that they are affected by quantization distortion. However, CMHMMs may not be suitable for noisy speech recognition because of the false assumption of a normal distribution. The DHMMs can represent more complicated shapes and they are expected to be useful for noisy speech.

Recently, discrete-mixture HMMs (DMHMMs), for which the quantization size can be reduced, have been proposed in [1,2]. DMHMMs require a smaller amount of training data than ordinary DHMMs. However, they still require a larger amount of training data than CMHMMs. To solve the problem of trainability, we proposed a MAP estimation of DMHMMs to further reduce the amount of training data [3]. It was reported that this method achieved an average error rate reduction of 28.1% in nonstationary conditions compared with CMHMM-based recognition [4].

In this paper, we propose a normalization method of DMHMMs based on histogram equalization (HEQ). This technique is commonly applied for feature space normalization [5]. In this method, a transform function is calculated directly from the histograms of both training data and test data, and the method can compensate the nonlinear effects of additive noise. This method can be applied to the normalization of an input feature vector. However, it cannot be used for model space normalization if CMHMMs are used as acoustic models. In a normal distribution, the shape of the distribution is represented by a continuous function that has two parameters, the mean and variance. The mean can be shifted by the transform function based on the HEQ method. However, the shape of the distribution cannot be modified by such a nonlinear transform function because it is determined only by the variance. In contrast, the shape of a DHMM can be modified because each sample value of the discrete stochastic variable can be shifted by the nonlinear transform function.

*e-mail: tkosaka@yz.yamagata-u.ac.jp
†e-mail: katoh@yz.yamagata-u.ac.jp
‡e-mail: kohda@yz.yamagata-u.ac.jp

In this paper, we propose both feature space and model space normalization based on the HEQ method. The normalization in model space has a merit compared with that in feature space. In the former, a transform function can be prepared for each acoustic model or model class separately. It is expected that the transform functions will depend on the feature of each phoneme or phoneme class. Then, experiments on normalization using multiple transform functions are conducted in this study to verify the effectiveness of multiple transformations.

The paper is organized as follows. In Sect. 2, we present a brief overview of the parameter estimation of DMHMMs. In Sect. 3, both model and feature space normalization methods based on HEQ are described. The experimental setup is described in Sect. 4. In Sect. 5, we present results showing the improved performance of this method. Finally, we conclude this paper in Sect. 6.

2. PARAMETER ESTIMATION OF DISCRETE-MIXTURE HMMS

2.1. Discrete-Mixture HMMs

In this section, DMHMMs are briefly introduced. In recent years, two types of DMHMM have been proposed. One is subvector-based quantization [2] and the other is scalar-based quantization [1]. In the former method, feature vectors are partitioned into subvectors, and then the subvectors are quantized using separate codebooks. In the latter, each dimension of the feature vectors is scalar-quantized. The quantization size can be reduced markedly by partitioning the feature vectors. For example, in [2], the quantization size was reported as 2 to 5 bits, and in [1], it was 4 to 6 bits. Because the quantization size is small, the DMHMM has superior trainability for acoustic modeling. In this work, subvector quantization is used because it has been reported that the subvector-based method is more effective than the scalar-quantized method.

The subvector-based method can be described as follows. The feature vector is partitioned into S subvectors, $\boldsymbol{o}_t = [\boldsymbol{o}_{1t}, \ldots, \boldsymbol{o}_{st}, \ldots, \boldsymbol{o}_{St}]$. VQ codebooks are provided for each subvector, and then the feature vector $\boldsymbol{o}_t$ is quantized,

$$q(\boldsymbol{o}_t) = [q_1(\boldsymbol{o}_{1t}), \ldots, q_s(\boldsymbol{o}_{st}), \ldots, q_S(\boldsymbol{o}_{St})]. \tag{1}$$

The output distribution of the DMHMM, $b_i(\boldsymbol{o}_t)$, is given by

$$b_i(\boldsymbol{o}_t) = \sum_m w_{im} \prod_s \hat{p}_{sim}(q_s(\boldsymbol{o}_{st})), \tag{2}$$

where w_{im} is the mixture coefficient for the mth mixture in state i, and $\hat{\boldsymbol{p}}_{sim}$ is the probability of the discrete symbol for the sth subvector.

2.2. MAP Estimation for DMHMM

In ML estimation, the effect of the prior distribution is ignored; however, an appropriate prior distribution is used for parameter estimation in MAP estimation. In this section, the training of DMHMMs based on MAP estimation is described. The ML estimate of the discrete probability $p_{sim}(k)$ is calculated in the following form:

$$p_{sim}(k) = \frac{\sum_{t=1}^{T} \gamma_{imt}\, \delta(q_s(\boldsymbol{o}_{st}), k)}{\sum_{t=1}^{T} \gamma_{imt}} \tag{3}$$

$$\delta(q_s(\boldsymbol{o}_{st}), k) = \begin{cases} 1 & q_s(\boldsymbol{o}_{st}) = k \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where k is the index of the subvector codebook and γ_{imt} is the probability of the mth mixture component being in state i at time t. We assume that the prior distribution can be represented by the Dirichlet density. The MAP estimate of DMHMM, $\hat{p}_{sim}(k)$, is given by

$$\hat{p}_{sim}(k) = \frac{\tau \cdot p^0_{sim}(k) + n_{im} \cdot p_{sim}(k)}{\tau + n_{im}} \tag{5}$$

$$n_{im} = \sum_{t=1}^{T} \gamma_{imt}, \tag{6}$$

where $p^0_{sim}(k)$ is the constrained prior parameter and τ indicates the relative balance between the corresponding prior parameter and the observed data. In our experiments, τ was set to 10.0 based on the results of experiments in [4]. Although both the mixture coefficient and the transition probability can be estimated by MAP, only the output probability is estimated by MAP in this paper.

2.3. Prior Distribution

The specification of the parameters of prior distributions is one of the key issues of MAP estimation. In our work, it is assumed that the prior distributions can be represented by models that are converted from CMHMMs to DMHMMs. In this case, the parameters of the prior distribution $p^0_{sim}(k)$ are given by

$$p^0_{sim}(k) = \frac{b'_{sim}(\boldsymbol{v}_s(k))}{\sum_k b'_{sim}(\boldsymbol{v}_s(k))}, \tag{7}$$

where $b'_{sim}(\ )$ is the probability density of the CMHMM, and $\boldsymbol{v}_s(k)$ is the centroid for each subvector s. While $p^0_{sim}(k)$ has the constraint that it must be a normal distribution, $\hat{p}_{sim}(k)$ in Eq. (5) does not have such a constraint. Thus, it is expected that $\hat{p}_{sim}(k)$ will be updated to represent more complicated shapes in the training session.

2.4. Compensation for Discrete Distributions

To improve noise robustness, a compensation method for discrete distributions is applied. It is more likely that the significant degradation of output probability will appear

in the case of mismatch conditions caused by unknown noise. This method can reduce the negative effect of unknown noise in the decoding process. It is particularly effective for short-duration noise [4]. The compensation method is given as follows: If $\hat{p}_{sim}(q_s(\boldsymbol{o}_{st})) < dth$ in Eq. (2), the output probability is set to dth, where dth is the threshold for the subvector.

3. CODEBOOK NORMALIZATION BASED ON HISTOGRAM EQUALIZATION

The HEQ technique is commonly applied for feature space normalization [5]. In this paper, this technique is applied for both feature space and model space normalization of DMHMMs. Model space normalization can be realized as a method of codebook normalization of DMHMMs. While model space normalization can be applied to DMHMMs, it cannot be applied to CMHMMs. For CMHMMs, the shape of the distribution cannot be modified by a nonlinear transform function because the shape is determined only by the variance. In contrast, the shape of the DHMM can be modified because each sample value of the discrete stochastic variable can be shifted using the nonlinear transform function. There is a merit in choosing model space normalization. In the method of model space normalization, a transform function can be prepared for each acoustic model or model class. For example, a voiced model and an unvoiced model can be normalized separately. It is expected that a more accurate method using multiple transformations will improve recognition performance.

The basic idea of HEQ is that a coefficient is transformed from one probability distribution to fit another. Cumulative density functions (CDFs) of both training and test data are used to calculate the transformation function. Two types of normalization, model space normalization and feature space normalization, are described. The transform function $HEQ_m(\ )$ for codebook normalization can be written as follows:

$$q'_s(\boldsymbol{o}_{st}) = HEQ_m(q_s(\boldsymbol{o}_{st})) = C_E^{-1}(C_T(q_s(\boldsymbol{o}_{st}))), \qquad (8)$$

where C_E is the CDF estimated from test data and C_T is the CDF from training data. Note that only the centroid q_s is transformed and the discrete probability $\hat{p}_{sim}$ is not changed. Then this normalization can be carried out using a small amount of input speech for CDF estimation. Furthermore, since all models share a set of codebooks, it is not necessary to normalize each model individually.

The transform function $HEQ_f(\ )$ for feature space normalization is given by

$$\boldsymbol{o}'_{st} = HEQ_f(\boldsymbol{o}_{st}) = C_T^{-1}(C_E(\boldsymbol{o}_{st})). \qquad (9)$$

In this case, both CDFs, C_T and C_E, are the same as the CDFs in Eq. (8), and $HEQ_f(\ )$ is the inverse transform of $HEQ_m(\ )$.

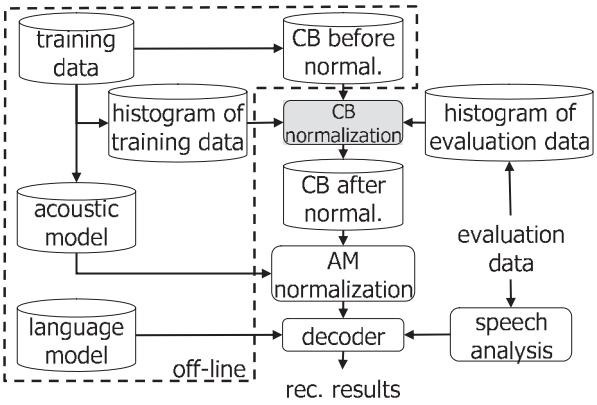

Fig. 1 Block diagram of codebook normalization.

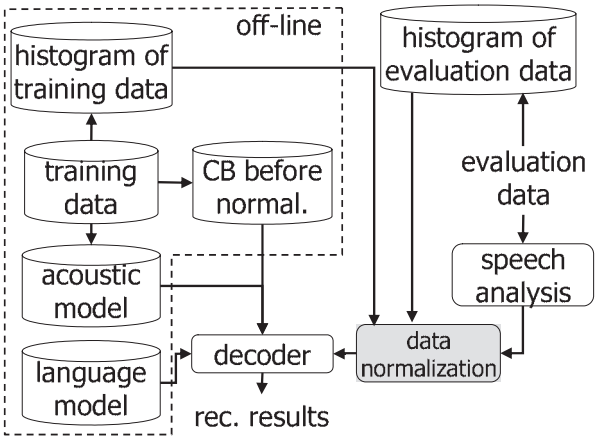

Fig. 2 Block diagram of feature vector normalization.

In the experiment, the parameters are 39 MFCCs with 12 mel cepstrum, log energy and their first- and second-order derivatives. There are various HEQ methods for transforming time derivatives [6]. In this work, each dimension is transformed independently.

Figure 1 shows the block diagram of codebook normalization, and Fig. 2 shows that of feature vector normalization. In Fig. 1, histograms derived from both training and test data are calculated to make a transform function. Then each codebook for the subvectors is normalized using the transform function. Only acoustic models are modified by using the normalized codebooks, and they are used in decoding process. In Fig. 2, each input utterance in the evaluation data is analyzed and normalized.

4. EXPERIMENTAL SETUP

The analysis conditions are summarized in Table 1. A set of shared state triphones was used as an acoustic model. The total number of states was 2,000, and the number of mixture components was 16.

For experiments, we used "JNAS: Japanese Newspaper

Table 1 Analysis conditions.

Sampling rate	16 kHz
Frame period	8 ms
Frame length	32 ms
Analysis	MFCC (1–12), log power+Δ + $\Delta\Delta$

Table 2 Codebook design.

Parameter	log P	c_1, c_2	c_3, c_4	c_5, c_6	c_7, c_8	c_9, c_{10}	c_{11}, c_{12}
Codebook size	64	64	64	64	64	64	64

Table 3 Set of recognition experiments.

Model	Normalization	Notation		
		w/o Normalization	Noise	Utterance
DMHMM	Feature	DMHMM	DMHMM-fer-noise	DMHMM-fer-utter
	Model		DMHMM-mod-noise	DMHMM-mod-utter
CMHMM	Feature	CMHMM	CMHMM-fer-noise	CMHMM-fer-utter
	Model		CMHMM-mod-noise	—

Article Sentences" as training and test data, which contains speech recordings and their orthographic transcriptions. Text sets for reading were extracted from the "Mainichi Shinbun" newspaper. We prepared two sets of training data. One was used for clean training, and the other was used for multicondition training [7]. The training data set consisted of 15,732 Japanese sentences uttered by 102 male speakers. For clean training, no noise was added to the data. For multicondition training, the utterances were divided into 20 subsets. No noise was added to 4 subsets. In the rest of the data, noise was artificially added. Four types of noise (car, exhibition hall, crowd and train) were selected and added to the utterances at SNRs of 20, 15, 10 and 5 dB. The noise data were selected from the JEIDA noise database [8], and the type of noise was determined with reference to the AURORA training set [7]. The set of clean training data was used for parameter estimation of the initial CMHMMs. After obtaining the initial CMHMMs, they were converted into DMHMMs by Eq. (7). In order to obtain noisy acoustic models, the parameters of DMHMMs were retrained using multicondition training data. Initial CMHMMs were also retrained using multicondition data, and retrained models were used for comparative experiments. The total number of states and the number of mixture components were the same as those of the DMHMM.

Two sets of test data were prepared for evaluation.

Testset A The noise condition is the same as that in the multicondition training set. Car, exhibition hall, crowd and train noises are used.

Testset B The noise condition is different from that in the multicondition training dataset. Station, factory, street crossing and elevator hall noises are used.

In testset A, the selected noise data were similar to the AURORA closed testset. With the exception of stationary noise data such as air conditioning noise and noise data in testset A, there were six types of noise. Four types of noise data out of the six were used as testset B, and the rest (distribution center and public telephone box) were omitted because they were similar to the noise data in testset B. Each type of noise was added to 100 sentences uttered by 10 male speakers at SNRs of 10 dB. Thus, the number of utterances was 400 for each testset.

All results presented here were extracted using a decoder with a word bigram with a 5 K word vocabulary. Table 2 shows the subvector allocation and codebook size. In the table, although Δ and Δ^2 are omitted, these codebooks were designed in the same manner. The codebook design was determined with reference to the results in [2] and the split vector quantizer in the DSR front end [9]. In [2], it was reported that DMHMMs with from 9 to 24 subvectors showed better performance. The feature vector was partitioned into subvectors that contain two consecutive coefficients. The consecutive coefficients that comprise subvectors are expected to be more correlated. Also, it was reported in [2] that subvectors that contained consecutive coefficients performed well. The LBG algorithm was utilized for creating the codebook. Multicondition training data was used for codebook creation.

5. RESULTS AND DISCUSSION

5.1. Comparison of Normalization Methods

Table 3 shows the set of experiments we conducted in this section. The nine methods shown in the table were compared in Japanese speech recognition experiments. The description of the table is as follows:

model Two types of model were compared. '*DMHMM*' means a discrete-mixture HMM trained by multicondition method. '*CMHMM*' is a conventional continuous-mixture HMM trained in the same way and is used for performance comparison.

normalization Two types of normalization were com-

pared. '*Feature*' means feature space normalization and '*model*' means model space normalization.

normalization data Two types of normalization data were compared. In the '*noise*' condition, the histogram of the test data was calculated for all the utterances of one noise type. In the '*utterance*' condition, the histogram of the test data was calculated only by each utterance that would be recognized. Then, histogram estimation is carried out using a very short calibration speech in this case.

For example, '*DMHMM-fer-noise*' means that the DMHMM is normalized in feature space using all the utterances of one noise type, and '*DMHMM*' means that the DMHMM is used without normalization. As we mentioned in Sect. 3, HEQ cannot be applied to CMHMMs in model space in a general way because the shape of the distribution is represented by the variance. In our experiments, model space normalization of the CMHMM was carried out by transforming mean values only. Thus, the shape of the distribution of the CMHMM was not changed in this case. The model space normalization of the CMHMM for each utterance was not performed. For DMHMMs, all acoustic models can be normalized by changing only the codebooks. In the case of CMHMMs, however, it is difficult to perform the normalization because the mean values of all the models should be normalized for every single utterance and the calculation cost is high. The threshold value *dth* was set to 2.5×10^{-4} in all the experiments in this section. A detailed discussion of the threshold is described in Sect. 5.3.

Figure 3 shows the recognition results (word error rate) for testset A and testset B, and the same results are shown in Table 4.

In the case of testset B, the recognition performance of the DMHMM was greater than that of the conventional CMHMM. All types of normalization methods provided significant improvements with respect to the DMHMM for testset B. The improvements of normalization for testset A were small. Because testset A consisted of known conditions and models were matched to noisy conditions, the improvements were not large. These results mean that the codebook normalization based on HEQ is able to compensate the nonlinear mismatch well.

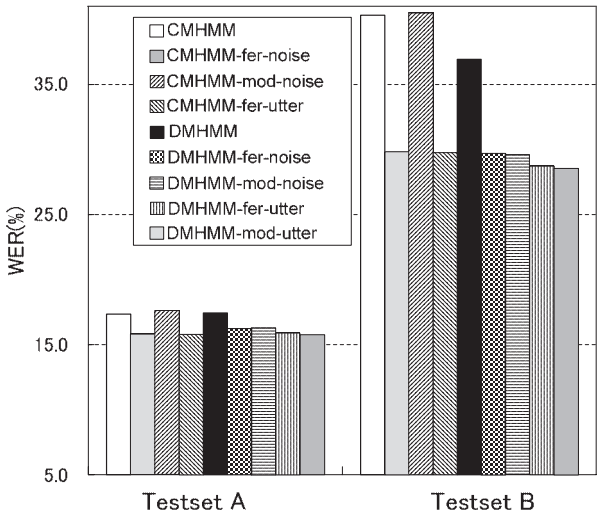

Fig. 3 Recognition results for testset A and testset B.

Table 4 Recognition results for testset A and testset B. Average WERs (%) of four types of noise are indicated.

	w/o norm.	-fer-noise	-mod-noise	-fer-utter	-mod-utter
testset A					
CMHMM	17.34	15.84	17.63	**15.82**	—
DMHMM	17.42	16.25	16.33	15.92	**15.76**
testset B					
CMHMM	40.32	29.82	40.53	**29.79**	—
DMHMM	36.93	29.71	29.61	28.78	**28.55**

The performance of HEQ in feature space was similar to that in model space for both testset A and testset B with the exception of '*CMHMM-mod-noise*.' This can be interpreted to mean that the direction of transformation is different between the two HEQ methods, but the effect is similar. In order to obtain further improvements for HEQ in model space, a transform function may be prepared for each acoustic model. The recognition performance of '*CMHMM-mod-noise*' was worse than that of the CMHMM without normalization. In the case of '*CMHMM-mod-noise*,' only mean values were normalized and the shape of the distribution was not changed. This fact means that the normalization of the distribution shape is important for HEQ in model space.

Comparing the difference in performance between '*noise*' and '*utterance*,' '*utterance*' showed slightly greater improvements on average. It is shown that one utterance is sufficient to estimate the transform functions. The average length of a test utterance is 3.9 s. Thus, the proposed normalization can be carried out using a very short calibration speech. In both testset A and testset B, the best performance was obtained for the '*DMHMM-mod-utter*' condition. However, the difference in performance between '*DMHMM-fer-utter*' and '*DMHMM-mod-utter*' was small. Hence, the superiority of model space normalization could not be confirmed in these experiments.

Figures 4 and 5 show the recognition results for each noise condition for testset A and testset B, respectively. From these results, it turns out that recognition performance depends on the noise type. In general, histogram normalization was effective for noise data in which the variation in noise was large, and it was less effective for noise data

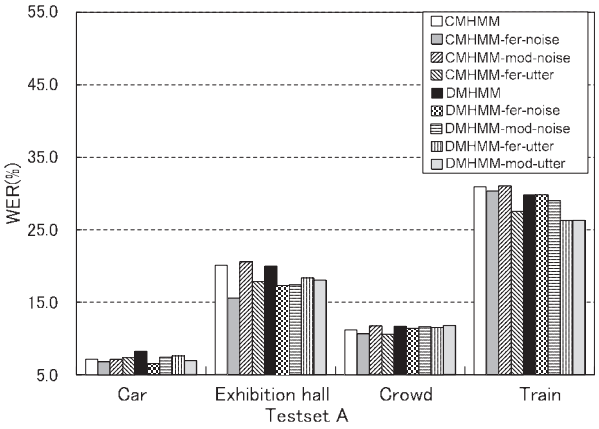

Fig. 4 Recognition results for each noise in testset A.

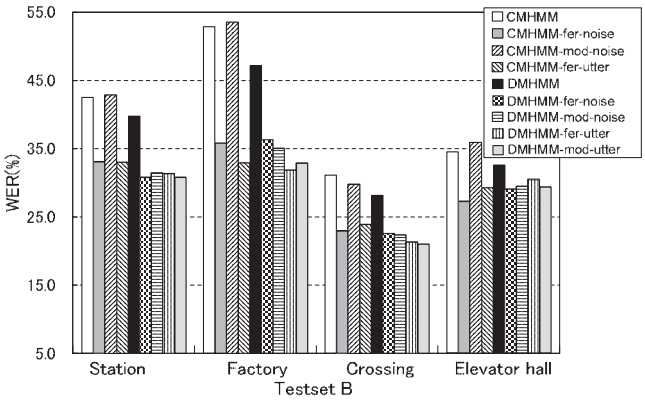

Fig. 5 Recognition results for each noise in testset B.

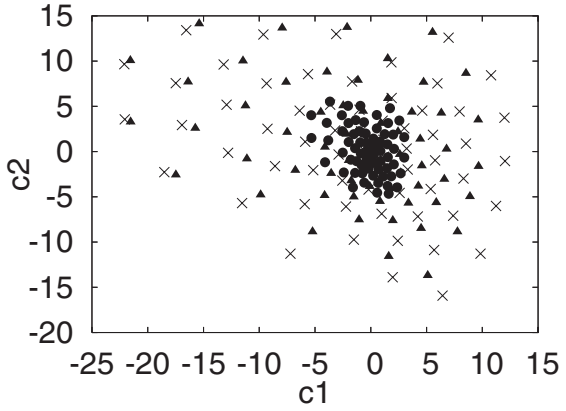

Fig. 6 Movement of codebook centroids after normalization for "train condition." ×: before normalization, ▲: normalization using entire evaluation data, ●: normalization using one utterance.

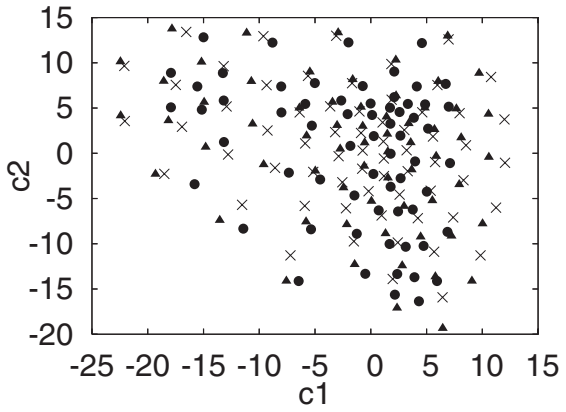

Fig. 7 Movement of codebook centroids after normalization for "car" condition. ×: before normalization, ▲: normalization using entire evaluation data, ●: normalization using one utterance.

with a small variation such as "crowd," "exhibition hall" and "elevator hall." For testset A, a significant improvement was obtained for "train" using one utterance compared with the '*noise*' condition, while other noise types were not. This is because large variations of the noise condition caused by passing train occur in the "train" condition. The "crossing" condition in testset B is similar to the "train" condition, because large variations of noise are observed due to cars passing. Figures 6 and 7 show the movement of codebook centroids on the C1–C2 plane after normalization. Figure 6 shows results for the "train" condition and Fig. 7 shows those for the "car" condition. Compared with the centroids before normalization for "train," the difference in position is not large in the case of normalization using the entire evaluation data. However, in the case of normalization using one utterance, these centroids move significantly. In the "car" condition, because the noise spectrum is concentrated at low frequency and is relatively stable, the movement of centroids is small.

5.2. Normalization Using Multiple Transform Functions

In the previous section, only one transform function was used for normalization. However, it is expected that the transform function depends on the features of each phoneme or phoneme class. For example, in the MLLR adaptation method [10], which is widely used as a method of model space transformation for speaker adaptation, multiple transform functions are usually used to improve adaptation performance. From a similar point of view, experiments on normalization using multiple transform functions were conducted. Since each transform function needs to be used for each model or model class in this method, model space normalization method is required. Normalization using multiple transform functions cannot be carried out by the conventional feature space approach.

In the experiments, acoustic models were phonetically classified into voiced and unvoiced groups. Those groups are shown in Table 5. Both silence (sil) and closure (cl) classes were classified into unvoiced classes. Since the distinction between voiced and unvoiced classes was based

Table 5 Classified list of phonemes.

Voiced	a aa i ii u uu e ee ei o oo ou b d g m n N z j w y xy r
Unvoiced	h f s sh ts ch p t k cl sil

Table 6 Comparison of recognition performance between 1-class and 2-class normalizations using DMHMM on testset B (WER %).

	w/o norm.	-fer-noise	-mod-noise 1-class	-mod-noise 2-class
Station	39.75	30.85	31.47	30.43
Factory	47.20	36.34	35.09	33.44
Crossing	28.16	22.57	22.36	22.26
Elevator hall	32.61	29.09	29.50	27.64
Ave.	36.93	29.71	29.61	28.44

on recognition results using acoustic models without normalization, some determination errors were included. For the normalization of the voiced class, the histogram was calculated using voiced speech and it was used for the normalization of voiced phonemes. For the unvoiced class, the histogram was calculated using voiced and unvoiced speech and it was used for the normalization of unvoiced phonemes because it showed better performance than the method by which the histogram was calculated using only unvoiced speech. Lack of data sometimes causes a problem in such a multiple classification approach. In particular, in the case of '*utter*,' lack of data tends to be a problem because the duration is short. Therefore, the experiments on testset B were conducted with the condition of '*noise*' as the first stage.

The results are shown in Table 6. From the results, the 2-class normalization method showed better results than the 1-class normalization method. Compared with feature space normalization (29.71%), 2-class normalization (28.44%) showed statistically significant improvement (significance level of 5%), while the performance of 1-class normalization did not show significant improvement. As described above, multiple transform normalization was successfully performed in model space and it showed better performance than feature space normalization.

5.3. Investigation on Likelihood Compensation

All experiments described above were performed with $dth = 2.5 \times 10^{-4}$ which was the threshold for compensation and introduced in Sect. 2.4. In order to clarify the robustness against the value of this threshold, recognition experiments were performed on testset B with various values of threshold. In the experiments, *dth* was varied from 0 to 1.0×10^{-3} and the word error rate for each

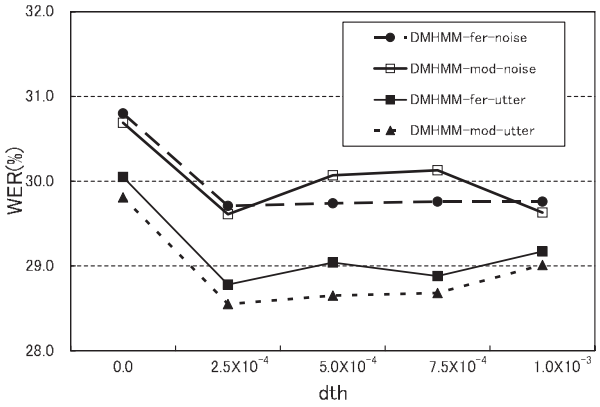

Fig. 8 Recognition performance at various threshold values of likelihood compensation.

method was calculated. The results are shown in Fig. 8. The performance of the methods using HEQ was relatively stable apart from the case of $dth = 0.0$. From these results, it can be concluded that the proposed method is robust against the variation in threshold.

6. CONCLUSIONS

In this paper, we proposed a normalization method of discrete-mixture HMMs (DMHMMs) with the aim of improving the performance of recognition under noisy conditions. The normalization method was based on histogram equalization (HEQ) and can compensate the nonlinear effects of additive noise. Both model space normalization and feature space normalization methods were proposed. It was difficult to apply the HEQ method to CMHMMs in model space in a general way, because the shape of the distribution was determined by the variance. In contrast, the codebook normalization of the DMHMM made model space normalization possible. In our experiments, the model space normalization of the CMHMM was carried out by transforming the mean values only. In this case, the shape of the distribution of the CMHMM was not changed and the recognition performance was unsatisfactory.

From the results of recognition experiments, both feature and model space normalization methods were effective for noise-robust speech recognition. From the comparison between feature and model space normalization, the recognition performance was similar when a single transform function was used. Model space normalization using multiple transform functions showed better performance than the method using a single transform function.

In this paper, only 2-class transform functions were used for the method using multiple transform functions. We plan to improve recognition performance further by testing a wide variety of phoneme classes. We also plan to use this method for speaker normalization in an LVCSR task such as "CSJ: Corpus of Spontaneous Japanese."

ACKNOWLEDGMENTS

The authors express their gratitude to Mr. D. Endo and Mr. Y. Saito of Yamagata University for their help in completing some of the experiments.

REFERENCES

[1] S. Takahashi, K. Aikawa and S. Sagayama, "Discrete mixture HMM," *Proc. ICASSP 1997*, pp. 971–974 (1997).

[2] S. Tsakalidis, V. Digalakis and L. Newmeyer, "Efficient speech recognition using subvector quantization and discrete-mixture HMMs," *Proc. ICASSP 1999*, pp. 569–572 (1999).

[3] T. Kosaka, M. Katoh and M. Kohda, "Noisy speech recognition with discrete-mixture HMMs based on MAP estimation," *Proc. ICA 2004*, Vol. II, pp. 1691–1694 (2004).

[4] T. Kosaka, M. Katoh and M. Kohda, "Robust speech recognition using discrete-mixture HMMs," *IEICE Trans. Inf. Syst.*, **E88-D**, 2811–2818 (2005).

[5] A. Torre, J. C. Segura, C. Benitez, A. M. Peinado and A. J. Rubio, "Non-linear transformations of the feature space for robust speech recognition," *Proc. ICASSP 2002*, pp. 401–404 (2002).

[6] Y. Obuchi, "Delta-cepstrum normalization for robust speech recognition," *Proc. ICA 2004*, pp. 2587–2590 (2004).

[7] D. Pearce and H.-G. Hirsch, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," *Proc. ICSLP 2000*, Vol. 4, pp. 29–32 (2000).

[8] Japanese Electronic Industry Development Association, "JEIDA noise database," http://www.sunrisemusic.co.jp/dataBase/fl/noisedata01_fl.html

[9] ETSI ES 201 108 V1.1.1, "STQ; distributed speech recognition; front-end feature extraction algorithm; compression algorithms," ETSI Standard (2000).

[10] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, **9**, 171–185 (1995).