A STUDY OF SWITCHING STATE SEGMENTATION IN SEGMENTAL SWITCHING LINEAR GAUSSIAN HIDDEN MARKOV MODELS FOR ROBUST SPEECH RECOGNITION

Donglai ZHU, Qiang HUO and Jian WU

Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong
(Email: dlzhu@cs.hku.hk, qhuo@cs.hku.hk, jianwu@microsoft.com)

ABSTRACT

In our previous works, a Switching Linear Gaussian Hidden Markov Model (SLGHMM) and its segmental derivative, SSLGHMM, were proposed to cast the problem of modeling a noisy speech utterance in robust automatic speech recognition by a well-designed dynamic Bayesian network. An important issue of SSLGHMM is how to specify a switching state value for each frame of feature vector in a given speech utterance. In this paper, we propose several approaches for addressing this issue and compare their performance on Aurora3 connected digit recognition tasks.

1. INTRODUCTION

A Switching Linear Gaussian Hidden Markov Model (SLGHMM), as shown in Fig. 1(a), was proposed in [9] to compensate for the nonstationary distortion that may exist in a speech utterance to be recognized. In [12], a variational approach has been proposed to solve the approximate maximum likelihood (ML) parameter learning and probabilistic inference problems for SLGHMMs. Unfortunately, it is not computationally feasible for automatic speech recognition (ASR) applications that require prompt response. Therefore, a Segmental SLGHMM (SSLGHMM hereinafter), as illustrated in Fig. 1(b), was proposed in [9].

In an SSLGHMM, several assumptions are made to simplify the model. Each switching state q_t is assumed to be independent of all switching states at other time instances. Switching states are treated as observations rather than hidden variables as in SLGHMM. The values of switching states are assigned by an appropriate pre-segmentation procedure. Therefore, an important issue of SSLGHMM is how to specify a switching state value for each frame of feature vector in a given speech utterance. For the convenience of reference, we refer hereinafter the above problem as *switching state segmentation*. We have investigated several approaches to address the above issue and conducted a series of experiments on Aurora3 connected digit recognition tasks [1, 2, 3, 4]. In the following, we first present our approaches in section 2. Section 3 will then detail the experiments and present and discuss their results. Finally, some conclusions are drawn in section 4.

2. SWITCHING STATE SEGMENTATION APPROACHES

The overall architecture of our approaches for switching state segmentation in SSLGHMM is shown in Fig. 2 and works as follows. Given the feature vector sequence of an input speech utterance $Y = \{y_1, y_2, \cdots, y_T\}$, the value of switching state, q_t, is determined in two stages: *condition labeling* and *frame labeling*. The *condition labeling* process uses the result of voice activity detection (VAD) that segments the input utterance into speech and non-speech segments [7]. In general, the input utterance Y can be classified as belonging to one of E conditions as follows:

$$e^* = \arg\max_e \prod_{t=1}^{T} p(y_t|\Lambda_e^{VAD(t)}) , \quad (1)$$

where $VAD(t)$ is the result (taking "S" for speech or "N" for non-speech) of VAD at time t, $\Lambda^S = \{\Lambda_e^S\}$ and $\Lambda^N = \{\Lambda_e^N\}$ are two sets of Gaussian mixture models (GMMs) designed for speech and non-speech segments respectively. After the condition labeling for the whole utterance, the *frame labeling* process will determine the value of switching state for each *speech* frame with the corresponding condition dependent *speech* GMM $\Lambda_{e^*}^S$ as follows:

$$q_t = \arg\max_q p(q|y_t, \Lambda_{e^*}^S) . \quad (2)$$

In Eq. (1), the total likelihood is a product of the likelihood of speech segments and that of non-speech segments. Apparently, there are other options in designing the condition labeling rule by adjusting contributions from speech and non-speech segments.

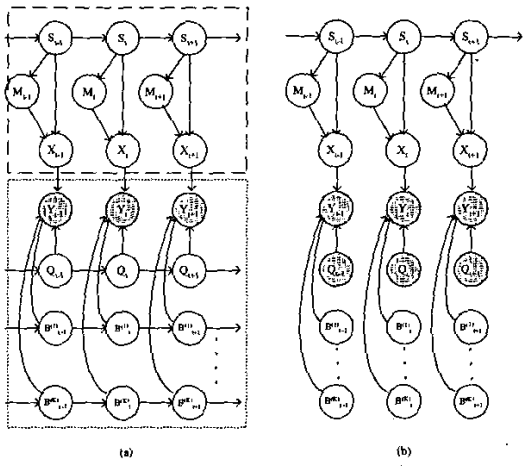

Fig. 1. Directed acyclic graph specifying conditional independence relations for (a) SLGHMM and (b) SSLGHMM

This research was supported by grants from the RGC of the Hong Kong SAR (Project Numbers HKU7022/00E and HKU7039/02E). Jian Wu is now with Microsoft Corporation, Redmond, USA.

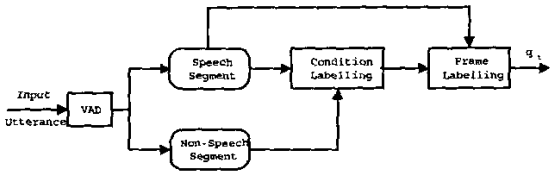

Fig. 2. Process of switching state segmentation in SSLGHMMs

Consequently, training procedures for Λ^S and Λ^N will also be different. In the following subsections, we detail five approaches we have investigated and refer to them hereinafter as A1, A2, A3, A4 and A5 respectively.

2.1. A1: Condition Labeling using both Speech and Non-Speech Segments

In this approach, the condition labeling is based on Eq. (1) using both speech and non-speech segments. The training procedure for two sets of GMMs, Λ^S and Λ^N is illustrated in Fig. 3 and explained in the following:

Step 1. Train a GMM with E Gaussian components by using non-speech segments of all training utterances. Each training utterance is then classified to one of the E classes that has the maximum likelihood of non-speech segments for the associated Gaussian component.

Step 2. Given the initial labeling of training utterances, for each condition class e, two GMMs, Λ_e^S and Λ_e^N are trained from speech segments and non-speech segments associated with class e respectively.

Step 3. Given $\{\Lambda_e^S\}$ and $\{\Lambda_e^N\}$, each training utterance is re-classified to the corresponding condition based on Eq. (1) using both speech and non-speech segments.

Step 4. Repeat steps 2 and 3 until no change of utterance labeling results or a maximum number of iterations is reached.

Each test utterance is also classified to the corresponding condition using Eq. (1).

2.2. A2: Condition Labeling using Speech Segments (1)

In this approach, the condition labeling is based on speech segments only and the labeling rule becomes

$$e^* = \arg\max_e \prod_{\substack{t=1 \\ VAD(t)=S}}^{T} p(y_t|\Lambda_e^S) \,. \tag{3}$$

The training procedure for the set of GMMs, Λ^S, is as follows:

Step 1. This step is the same as Step 1 in Approach 1.

Step 2. Given the initial labeling of training utterances, for each condition class e, a GMM Λ_e^S is trained with all speech segments associated with class e.

Step 3. Given $\{\Lambda_e^S\}$, each training utterance is re-classified to the corresponding condition based on Eq. (3) using speech segments only.

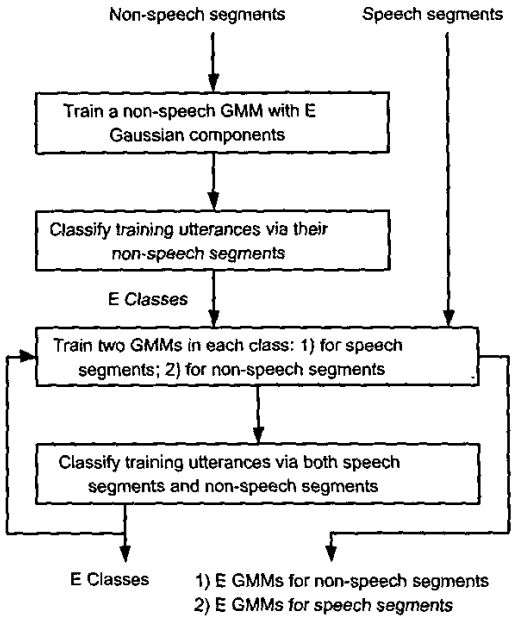

Fig. 3. Training of two sets of GMMs for speech and non-speech segments in Approach 1.

Step 4. Repeat steps 2 and 3 until no change of utterance labeling results or a maximum number of iterations is reached.

Each test utterance is also classified to the corresponding condition using Eq. (3).

2.3. A3: Condition Labeling using Speech Segments (2)

This approach is similar to Approach 2. The condition labeling rule for a test utterance is the same as in Eq. (3), but the training procedure for the set of GMMs, Λ^S, is different. The difference lies in the first step for the initial labeling of training utterances. Step 1 of Approach 3 read as

Step 1. Train a GMM with E Gaussian components by using speech segments of all training utterances. Each training utterance is then classified to one of the E classes that has the maximum likelihood of speech segments for the associated Gaussian component.

The other three steps are the same as in Approach 2.

2.4. A4: Condition Labeling using Non-Speech Segments

In this approach, the condition labeling is based on non-speech segments only and the labeling rule is as follows:

$$e^* = \arg\max_e \prod_{\substack{t=1 \\ VAD(t)=N}}^{T} p(y_t|\Gamma_e^N) \,, \tag{4}$$

where Γ_e^N represents the Gaussian component of condition e as trained in Step 1 of Approach 1. Each test utterance is also classified to the corresponding condition using Eq. (4).

2.5. A5: Condition Labeling using Speech Segments in Testing and Non-Speech Segments in Training

In this approach, each test utterance is classified to the corresponding condition according to Eq. (3) by using speech segments only. However, the set of speech GMMs $\{\Lambda_c^S\}$ are trained by using the first two steps of the training procedure in Approach 2. This approach was proposed in [5] and adopted in [10].

3. EXPERIMENTS AND RESULTS

3.1. Experimental Setup

We use Aurora3 database [1, 2, 3, 4] to verify our algorithm. Aurora3 contains utterances of connected digits in four European languages, namely Finnish, Spanish, German and Danish. All utterances were recorded by using both close-talking (CT) and hands-free (HF) microphones in cars under several driving conditions to reflect some realistic scenarios for typical in-vehicle ASR applications. For each language, the database is divided into three subsets according to matching degree between training data and test data: Well-Matched condition (WM), Middle Mismatched condition (MM) and High Mismatched condition (HM).

In our experiments, two front-ends are used for feature extraction from a speech utterance. The first front-end is the ETSI Advanced Front-End (AFE) as described in [7]. A feature vector sequence is extracted from the input speech utterance via a sequence of processing modules that include noise reduction, waveform processing, cepstrum calculation, blind equalization, and "server feature processing". Each frame of feature vector has 39 features that consists of 12 MFCCs (C_1 to C_{12}), a combined log energy and C_0 term, and their first and second order derivatives. The second front-end used in our experiments is called *basic front-end* (BFE). A feature vector sequence is extracted from the input speech utterance via two processing modules only, namely cepstrum calculation and "server feature processing" as described in [7]. In comparison with the ETSI Front-End described in [6], the main differences are, in our *basic front-end*, 1) no offset compensation, 2) the value of pre-emphasis coefficient is 0.9 instead of 0.97, 3) a power spectrum instead of a magnitude spectrum is used in filterbank integration. For both AFE and BFE, all the feature vectors are computed from a given speech utterance, but the feature vectors that are sent to the speech recognizer and the training module of SSLGHMMs are those corresponding to speech frames, as detected by a VAD module described in Annex A of [7]. This also explains why only *speech frame labeling* is concerned in Eq. (2).

Each digit is modeled as a whole word left-to-right SSLGHMM with 16 emitting states, 3 Gaussian mixture components with diagonal covariance matrices per state. Besides, two pause models, "sil" and "sp", are created to model the silence before/after the digit string and the short pause between any two digits. The "sil" model is a 3-emitting state SSLGHMM with a flexible transition structure as that of HMM described in [8]. Each state is modelled by a mixture of 6 Gaussian components with diagonal covariance matrices. The "sp" model consists of 2 dummy states and a single emitting state which is tied with the middle state of "sil".

During recognition, an utterance can be modelled by any sequence of digits with the possibility of a "sil" model at the beginning and at the end and a "sp" model between any two digits. All of the recognition experiments are performed with the search engine of HTK3.0 toolkit [13].

Table 1. Number of training utterances classified to different conditions using different condition labeling approaches (AFE, WM condition, Finnish language)

Condition	A1	A2	A3	A4/A5
1	10	157	378	9
2	288	710	482	217
3	167	329	7	366
4	491	416	714	754
5	634	240	250	32
6	514	444	348	727
7	233	346	396	500
8	743	438	505	475

Table 2. Number of training utterances classified to different conditions using different condition labeling approaches (BFE, WM condition, Finnish language)

Condition	A1	A2	A3	A4/A5
1	353	300	391	468
2	169	506	345	180
3	638	786	435	813
4	304	243	694	238
5	250	245	5	253
6	235	218	423	265
7	651	517	18	385
8	480	605	789	675

In condition labeling, the number of conditions is set to 8, i.e., each input utterance is labeled to one of 8 conditions. In frame labeling, the number of classes in each condition is set to 32, i.e., each speech frame in an utterance is classified to one of 32 classes in the corresponding condition. Therefore there are $8 \times 32 = 256$ linear Gaussian dynamic streams in the SSLGHMM. In the approaches where training utterances are required to be iteratively re-classified (i.e., approaches 1, 2, 3), ten iterations are performed.

In ML training of SSLGHMMs, an important implementation issue is the specification of initial values for model parameters. The initial values for continuous density HMM (CDHMM) parameters are obtained by running the training scripts published in Aurora3 CDs, i.e., the standard ML training implemented in HTK. The initial values for "bias means" are obtained in the following two steps: Firstly, the values are set to zeros. Secondly, the initial values are estimated by performing one EM iteration in the second step of an ML training procedure for a stochastic vector mapping (SVM) based approach as described in [10, 12]. The initial values for "bias variances" are set to a small value 0.001. Starting from the above initial values, five EM iterations are performed. After each iteration, the parameters for CDHMMs and switching linear Gaussians are both updated. In our current experiments, because a small initial value is set for bias variances, bias means and variances converge very slowly in the ML training process. After five EM iterations, values of most bias means and variances remain unchanged or have changed only slightly. This makes the performance of ML-trained SSLGHMM system essentially the same as that of SVM-based system trained as follows: The same initial bias values and CDHMMs as in SSLGHMMs are used and 5 EM iterations are performed to update CDHMM parameters only by using SVM-normalized training data.

Starting from the above ML-trained SSLGHMMs, a series of

Table 3. Word error rate (in %) of ML-trained SSLGHMMs with different condition labeling approaches under WM condition of Finnish language

Font-Ends	A1	A2	A3	A4	A5
AFE	3.44	3.36	3.47	3.56	3.47
BFE	6.35	6.09	6.37	6.26	6.25

experiments are also performed for MCE training of SSLGHMM [11]. Five epoches are performed in the MCE training. Some control parameters are set as follows: $\alpha = 0.05$, $\beta = 0.0$, $\eta = 0.05$, $N = 8$ in "top-N list". Readers are referred to [11] for the meaning of the above control parameters as well as the details of the MCE training algorithm for SSLGHMMs.

3.2. Results of Different Condition Labeling Approaches

Tables 1 and 2 summarize for AFE and BFE front-ends respectively, the distribution of training utterances of Finnish language under WM condition by using different condition labeling approaches. Approaches 4 and 5 have the same utterance distribution because the training utterances are labeled using the same procedure with non-speech segments in both cases. We observed that different clustering results are obtained by using different condition labeling approaches and front-ends. Approach 2 leads to a more uniform distribution of training utterances among different conditions. Table 3 compares recognition results (word error rate in %) of ML-trained SSLGHMMs with five condition labeling approaches for both AFE and BFE. It is observed that Approach 2 achieves the best performance for both front-ends. This is the approach we used in [11] as well as in a benchmark evaluation on Aurora3 reported in the next subsection.

3.3. Evaluation Results of SSLGHMMs on Aurora3

We evaluated SSLGHMMs with the condition labeling Approach 2 on all four languages in Aurora3. For each language, experiments are performed only for well-matched condition. Both ML and MCE training are evaluated for both SSLGHMMs and traditional CDHMMs. Tables 4 and 5 summarize word error rates of different experiments for AFE and BFE respectively. In the above tables, "ML-CDHMM" and "MCE-CDHMM" refer to two baseline systems using traditional CDHMMs that have the same model structure as their CDHMM counterparts in SSLGHMMs, and trained under ML and MCE criteria respectively. It is observed that SSLGHMM systems achieve better performance than the CDHMM baseline systems when both are trained with the same criterion. For both SSLGHMM and CDHMM baseline systems, models trained with MCE criterion achieve better performance than those trained with ML criterion.

4. SUMMARY

In this paper, several approaches are proposed and compared empirically for switching state segmentation in SSLGHMM. Among them, the second approach achieves the best performance in different experiments. Using this approach, a benchmark evaluation is performed on Aurora3 tasks and we demonstrate again that the SSLGHMM approach achieves a state-of-the-art performance among results reported in literature on this task.

Table 4. Aurora3 Word Error Rate (in %) on WM Condition: AFE

	Fin.	Span.	Ger.	Dan.	Ave.
ML-CDHMM	3.95	3.39	4.87	6.02	4.56
ML-SSLGHMM	3.36	3.17	4.49	5.73	4.19
MCE-CDHMM	2.82	2.84	4.69	5.27	3.91
MCE-SSLGHMM	2.30	2.73	4.19	5.15	3.59

Table 5. Aurora3 Word Error Rate (in %) on WM Condition: BFE

	Fin.	Span.	Ger.	Dan.	Ave.
ML-CDHMM	7.50	4.58	7.25	9.17	7.13
ML-SSLGHMM	6.09	4.11	6.57	8.45	6.31
MCE-CDHMM	4.78	3.72	6.53	7.85	5.72
MCE-SSLGHMM	3.88	3.89	5.81	7.52	5.28

5. REFERENCES

[1] Aurora document AU/217/99, "Availability of Finnish SpeechDat-Car database for ETSI STQ WI008 front-end standardisation," Nokia, Nov 1999.

[2] Aurora document AU/271/00, "Spanish SDC-Aurora database for ETSI STQ Aurora WI008 advanced DSR front-end evaluation: description and baseline results," UPC, Nov 2000.

[3] Aurora document AU/273/00, "Description and baseline results for the subset of the SpeechDat-Car German database used for ETSI STQ Aurora WI008 Advanced DSR Front-end Evaluation," Texas Instruments, Dec 2001.

[4] Aurora document AU/378/01, "Danish SpeechDat-Car digits database for ETSI STQ-Aurora advanced DSR," Aalborg University, Jan 2001.

[5] J. Droppo, L. Deng and A. Acero, "Evaluation of SPLICE on the AURORA 2 and 3 Tasks," *Proc. ICSLP-2002*, 2002, pp.29-32.

[6] ETSI standard document, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front-end feature extraction algorithm; compression algorithms", ETSI ES 201 108 v1.1.3 (2003-09), 2003.

[7] ETSI standard document, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms", ETSI ES 202 050 v1.1.1 (2002-10), 2002.

[8] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," *ISCA ITRW ASR-2000*, Paris, France, September 2000.

[9] J. Wu and Q. Huo, "A switching linear Gaussian hidden Markov model and its application to nonstationary noise compensation for robust speech recognition," *Proc. Eurospeech-2003*, 2003, pp.977-980.

[10] J. Wu and Q. Huo, "Several HKU approaches for robust speech recognition and their evaluation on Aurora connected digit recognition tasks," *Proc. Eurospeech-2003*, 2003, pp.21-24.

[11] J. Wu, D. Zhu and Q. Huo, "A study of minimum classification error training for segmental switching linear Gaussian hidden Markov models," *Proc. ICSLP-2004*, 2004.

[12] J. Wu, "Discriminative speaker adaptation and environmental robustness in automatic speech recognition", Ph.D thesis, the Department of Computer Science, The University of Hong Kong, July 2004.

[13] S. Young and et al., *The HTK Book (for HTK V3.0)*, July 2000.