A Study on Hidden Structural Model and Its Application to Labeling Sequences

Yu Qiao [#1], Masayuki Suzuki [#2], and Nobuaki Minematsu [#1]

[#1] *Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan*
[#2] *Graduate School of Engineering, The University of Tokyo, Tokyo, Japan*
{qiao,suzuki,mine}@gavo.t.u-tokyo.ac.jp

***Abstract*—This paper proposes Hidden Structure Model (HSM) for statistical modeling of sequence data. The HSM generalizes our previous proposal on structural representation by introducing hidden states and probabilistic models. Compared with the previous structural representation, HSM not only can solve the problem of misalignment of events, but also can conduct structure-based decoding, which allows us to apply HSM to general speech recognition tasks. Different from HMM, HSM accounts for the probability of both locally absolute and globally contrastive features. This paper focuses on the fundamental formulation and theories of HSM. We also develop methods for the problems of state inference, probability calculation and parameter estimation of HSM. Especially, we show that the state inference of HSM can be reduced to a quadratic programming problem. We carry out two experiments to examine the performance of HSM on labeling sequences. The first experiment tests HSM by using artificially transformed sequences, and the second experiment is based on a Japanese corpus of connected vowel utterances. The experimental results demonstrate the effectiveness of HSM.**

I. Introduction

One of the major challenging problems in speech engineering is to deal with non-linguistic variations contained in speech signals. These variations are caused by the difference of speakers, communication channels, environment noise, etc. To overcome this difficulty, modern speech recognition approaches mainly make use of statistical methods (such as GMM and HMM) to model the distributions of acoustic features. These methods always require a large amount of data for training and can achieve relatively high recognition rates when there is a good match between training and testing. But it is well-known that the performance of speech recognizers drops significantly if mismatch exists. Let us consider children's spoken language acquisition. A child does not need to hear the voices of thousands of speakers before he (or she) can understand speech. This fact largely indicates that there may exist robust representations of speech that are nearly invariant to non-linguistic variations. We consider it is by these robust representations that children can acquire spoken language with very speaker-biased training data from their parents.

In our previous work [1], the third author proposed an invariant structural representation of speech which aims at removing the non-linguistic factors from speech signals. Different from classical speech models, the structural representations make use of globally contrastive features to model the global and dynamic aspects of speech and discard the local and static features. It can be proved that these contrastive features (f-divergence) are invariant to any invertible transformations and thus are robust to non-linguistic variations [2]. It is noted that our contrastive features are different from delta (or delta-delta) features often used in speech engineering. The delta features describe differential information of cepstrums and are not invariant to transformations. We have already demonstrated the effectiveness of this representation in ASR [3], [4], speech synthesis [5], and CALL [6].

However, the structural representation also has its limitations. A speech structure is constructed for every sequence independently. So there may exist misalignment between event sequences with the same linguistic contents. Moreover, structural representations don't have label information for events, which makes them difficult to be used for general speech recognition. To overcome these difficulties, this paper proposes Hidden Structure Model (HSM) by introducing hidden states and probabilistic analysis previous. Compared with the previous structural representation, HSM unifies structure construction and structure comparison into a single framework, and avoids the misalignment of events. Moreover, the introduction of hidden states allows HSM to conduct structure-based decoding. This further allows us to apply HSM to general phoneme recognition other than word recognition. HSM is similar to HMM in a sense that both make use of hidden states, but different from HMM in a sense that HSM contains the probability models of both locally absolute and globally contrastive features. This paper proposes the fundamental formulation of HSM and develops the algorithms for state inference, probability calculation and parameter estimation of HSM. We carry out two experiments on artificial data and connected Japanese vowel utterances. The experimental results exhibit that the combination of both absolute and contrastive features in HSM can improve the recognition rates.

II. Review of previous structural representations

This section gives a brief overview on the invariant structure theory and how to calculate structural representations from utterances. More details can be found in our previous works [1], [2], [4].

An invariant structure is constructed from a set of distributions. As preparation, we introduce an invariant metric between distributions. Consider two distributions $p_i(x)$ and $p_j(x)$ in feature space X. Assume there is an invertible

transformation $h : X \rightarrow Y$ (linear or nonlinear) which maps x into y. In this way, distributions $p_i(x)$ and $p_j(x)$ are converted in to $q_i(y)$ and $q_j(y)$. Under transformation h, $p(x)dx = q(y)dy$ and $dy = |\Phi(x)|dx$, where $\Phi(x)$ denotes the determinant of the Jacobian matrix of h. Thus we have $q(y) = q(h(x)) = p(x)|\Phi(x)|^{-1}$. Consider f-divergence [7] defined as

$$D_f(p_i, p_j) = \oint p_j(x) f\left(\frac{p_i(x)}{p_j(x)}\right) dx, \qquad (1)$$

where $f : (0, \infty) \rightarrow R$ is a real convex function and $f(1) = 0$. It can be proved that f-divergence is invariant to transformation: $D_f(q_i, q_j) = D_f(p_i, p_j)$ [2]. Moreover, we found that all the invariant integration measures $\oint M(p_i, p_j)dx$ must be in the form of f-divergence [2].

Consider feature space X and pattern P in X. Suppose P is composed of a sequence of K events $\{p_i\}_{i=1}^K$, where each event is described as distribution $p_i(x)$ in X. Note x can have multiple dimensions. Under h, pattern P in X is transformed to pattern Q in Y, and event $p_i(x)$ is converted to event $q_i(y)$. From pattern P, we can calculate a $K \times K$ divergence matrix $\mathcal{D}^P$ with $\mathcal{D}^P(i, j) = D_f(p_i, p_j)$ and $\mathcal{D}^P(i, i) = 0$. Here $\mathcal{D}^P$ provides a structural representation of pattern P. Similarly, we can obtain structural representation $\mathcal{D}^Q$ for pattern Q. Due to the invariance of f-divergence, $\mathcal{D}^Q \equiv \mathcal{D}^P$, and the structural representation is invariant to transformations.

In the next, we show how to calculate a structural representation from an utterance. As shown in Fig. 1, at first, we calculate a sequence of cepstrum from input speech waveforms. Then an HMM is trained from a single cepstrum sequence and each state of the HMM is regarded as an event. Thirdly we calculate f-divergences between each event pair. These divergences will form a distance matrix with zero diagonal, which can be seen as the structural representation. For convenience, we can expand its upper triangle into a structure vector if the f-divergence used is symmetric. It is easy to see that this structural representation must be invariant to transformations in feature space. In speech engineering, non-linguistic speech variations are often modeled as transformation of cepstrum feature space. Microphone and environment distortion modifies cepstrum features with an additive vector. And vocal tract length difference is often modeled as linear transformation of the cepstrum features [8]. With structural representation, the speech recognition can be seen as a structure matching problem, where the matching score of two structures $\mathcal{D}^P$ and $\mathcal{D}^Q$ is given by $D(P, Q) = \sum_{i,j} |\mathcal{D}^P(i, j) - \mathcal{D}^Q(i, j)|^2$.

III. Hidden Structure Model

In the previous structural representation, a distribution (event) sequence is calculated for each utterance independently of other utterances. There may exist misalignment between different distribution sequences. For example, let $P = \{p_1, p_2, ...\}$ and $Q = \{q_1, q_2, ...\}$ denote two distribution sequences calculated from two utterances of the same word 'aiueo'. Assume that p_3 of P comes from 'i', but q_3 of Q may come from 'u'. Another limitation of the structural

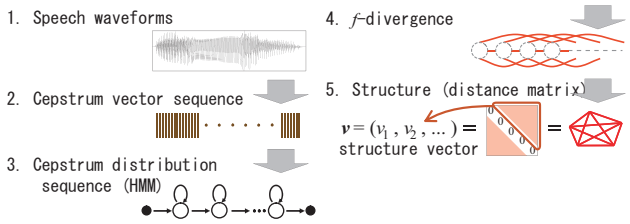

Fig. 1. Framework of structure construction.

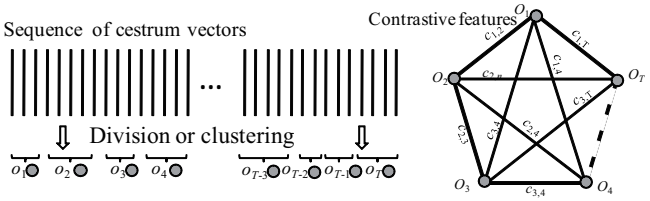

Fig. 2. Preprocessing of cepstrum sequence.

representation is that it doesn't include any label or category information of each event. Although the word recognition problem can be reduced to structure matching, it is difficult to extend this technique for other general speech recognition tasks, such as phoneme recognition. Moreover, in practice, contrastive features (f-divergences) are not strictly invariant due to noise and speaking styles. We need to consider a probabilistic model for contrastive features.

We notice that HMM doesn't have the above limitations. HMM avoids the misalignment problem by using DP-matching to align a cepstrum sequence with a sequence of HMM distributions. Moreover, HMM includes hidden states and has a flexible algorithm (Viterbi decoding) to estimate the most probable hidden state sequence for an observed sequence. This makes HMM suitable for solving general recognition tasks. Recall that the main advantage of the structural representation is that it makes use of contrastive features, which are robust to speaker difference. Inspired by these facts, we develop Hidden Structure Model for sequence data, which aims at combining contrastive features with a flexible and probabilistic model. Like HMM, HSM introduces the hidden states of observations and takes account for the labels of these hidden states. Unlike HMM, HSM models the distributions of both absolute and contrastive features that make it more robust to speaker differences.

A. Preprocessing of speech sequences for HSM

The contrastive features have to be calculated from events (sub-sequences or segments). For this reason, we need to divide a sequence $X = x_1, x_2, ..., x_M$ into a set of segments $O = o_1, o_2, ..., o_T$ with a preprocessing step (Fig. 2). Generally, we can use agglomerative clustering algorithm (ACA) [9] or HMM-based decomposition for sequence division [3], [4]. If we use ACA, each segment is a subsequence, denoted by, $o_t = x_{m_t}, x_{m_t+1}, ..., x_{e_t}$. If we use the second method, each segment is modeled as a Gaussian distribution $N(\bar{o}_t, V_t)$. For each segment pair o_{t_1} and o_{t_2}, we use c_{t_1,t_2} to denote the

contrastive feature between them.

B. Introduction of Hidden Structure Model

Generally speaking, HSM is a probabilistic model for sequence data, which takes account for joint distribution of both absolute and contrastive features. To begin with, we formally define the elements of HSM as the following.

1) N, the number of hidden states in HSM. We denote the set of individual states as $S = \{s_n\}_{n=1}^{N}$. We use q_t ($q_t \in S$) to represent the hidden state of o_t in sequence O. Then the state sequence is denoted by $Q = q_1, q_2, ..., q_T$.

2) State transition probability distribution $B = \{b_{i,j}\}$, where $b_{i,j} = p(q_{t+1} = s_j | q_t = s_i)$ $(1 \le i, j \le N)$ and $\sum_j b_{i,j} = 1$.

3) Initial state distribution $\pi = \{\pi_i\}$, where $\pi_i = p(q_1 = s_i)$ $(1 \le i \le N)$ and $\sum_i \pi_i = 1$.

4) Absolute observation probability (AOP) distribution in state j, $p(o_t | q_t = s_j)$. If the segment is a subsequence, we can calculate its mean as $\bar{o}_t = \frac{1}{e_t - m_t + 1} \sum_{i=m_t}^{e_t} x_i$. We assume that AOP has a Gaussian form,

$$p(\bar{o}_t | q_t = s_j) = N(\bar{o}_t | \mu_j^a, \Sigma_j^a). \tag{2}$$

Let $A = \{\mu_j^a, \Sigma_j^a\}$ denote the set of AOP parameters, and O_A the set of absolute features of sequence O.

5) Contrastive observation probability (COP) distribution for state i and state j, $p(c_{t_1,t_2} | q_{t_1} = s_i, q_{t_2} = s_j)$, where c_{t_1,t_2} represents the contrastive features (BD, KL-div. [4], [2]) between o_{t_1} and o_{t_2}. COP is assumed to have a Gaussian form,

$$p(c_{t_1,t_2} | q_{t_1} = s_i, q_{t_2} = s_j) = N(c_{t_1,t_2} | \mu_{i,j}^c, \Sigma_{i,j}^c). \tag{3}$$

Let $C = \{\mu_{i,j}^c, \Sigma_{i,j}^c\}$ denote the set of COP parameters, and O_C the set of contrastive features of sequence O.

One can see that items 1)-4) are the same as those of classical HMM, but item 5) is new, which describes the distribution of contrastive features. For convenience, we use a compact notation $\lambda = (A, B, C, \pi)$ to represent the complete model parameters.

Consider model λ, speech sequence $O = o_1, o_2, ..., o_T$ and its state sequence $Q = q_1, q_2, ..., q_T$. HSM calculates the joint probability of absolute features O_A and relative features O_C given model λ and state sequence Q as,

$$p(O|Q,\lambda) = p(O_A, O_C | Q, \lambda) = \frac{1}{\mathcal{N}(Q,\lambda)} \underbrace{\prod_{t=1}^{T} p(\bar{o}_t | q_t)}_{\text{Absolute part}} \underbrace{\prod_{1 \le t_1, t_2 \le T} p(c_{t_1,t_2} | q_{t_1}, q_{t_2})}_{\text{Contrastive part}}. \tag{4}$$

In general cases, normalization factor $\mathcal{N}(Q,\lambda)$ is required in Eq. 4 to ensure $\int_O p(O|Q,\lambda) = 1$. For simplicity, in this paper, we assume the independence of O_A and O_C, and the normalization factor reduces to 1. An example of HSM is depicted in Fig. 3. Note if we remove the contrastive part of Eq. 4, this probability calculation will be the same as that of HMM. On the other hand, if we remove the absolute part, Eq. 4 reduces to a probabilistic model of structural representation.

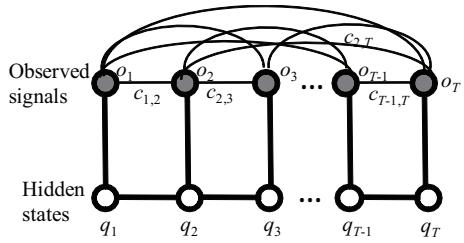

Fig. 3. An example of HSM. (HMM contains only the thick lines.)

We introduce the following variables $Z = \{z_{i,t}\}$, where

$$z_{i,t} = \begin{cases} 1 & \text{if } q_t = s_i \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that Z has the same information as Q. With $z_{i,t}$, we can rewrite Eq. 4 into

$$p(O|Q,\lambda) = p(O|Z,\lambda) = \prod_{t=1}^{T} \prod_{i=1}^{N} p(\bar{o}_t | s_i)^{z_{i,t}} \prod_{1 \le t_1, t_2 \le T} \prod_{i=1}^{N} \prod_{j=1}^{N} p(c_{t_1,t_2} | s_i, s_j)^{z_{i,t_1} z_{j,t_2}}. \tag{5}$$

Like in HMM, the probability of state sequence is given by

$$p(Q|\lambda) = p(Z|\lambda) = p(q_1) \prod_{t=2}^{T} p(q_t | q_{t-1}) = \prod_{i=1}^{N} p(s_i)^{z_{i,1}} \prod_{t=2}^{T} \prod_{i=1}^{N} \prod_{j=1}^{N} p(s_i | s_j)^{z_{i,t} z_{j,t-1}}. \tag{6}$$

Therefore, we have

$$p(O,Q|\lambda) = p(O,Z|\lambda) = p(O|Z,\lambda) p(Z|\lambda) = \prod_{i=1}^{N} p(s_i)^{z_{i,1}} \prod_{t=1}^{T} \prod_{i=1}^{N} p(\bar{o}_t | s_i)^{z_{i,t}} \prod_{t=2}^{T} \prod_{i=1}^{N} \prod_{j=1}^{N} p(s_i | s_j)^{z_{i,t} z_{j,t-1}} \prod_{1 \le t_1, t_2 \le T} \prod_{i=1}^{N} \prod_{j=1}^{N} p(c_{t_1,t_2} | s_i, s_j)^{z_{i,t_1} z_{j,t_2}}. \tag{7}$$

Calculate the log of the above equation,

$$\log p(O,Z|\lambda) = \sum_{i=1}^{N} z_{i,1} \log \pi_i + \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} z_{i,t} z_{j,t-1} \log b_{i,j} + \sum_{t=1}^{T} \sum_{i=1}^{N} \zeta_{i,t} z_{i,t} + \sum_{1 \le t_1, t_2 \le T} \sum_{i=1}^{N} \sum_{j=1}^{N} \eta_{i,j,t_1,t_2} z_{i,t_1} z_{j,t_2}. \tag{8}$$

where $\zeta_{i,t} = \log p(\bar{o}_t | s_i)$ and $\eta_{i,j,t_1,t_2} = \log p(c_{t_1,t_2} | s_i, s_j)$.

In the next, we introduce methods to solve the three problems of HSM, namely, state inference, probability calculation and parameter estimation.

C. State inference

Given model λ and observed stream O, the objective of state inference is to determine Z maximizing the following conditional probability,

$$\arg\max_{Z} p(Z|O,\lambda). \tag{9}$$

Using Bayesian theory, we have

$$p(Z|O,\lambda) = \frac{p(O,Z|\lambda)}{p(O|\lambda)} \propto p(O,Z|\lambda). \tag{10}$$

Thus the problem can be reduced to find Z which maximizes Eq. 8, $\max_Z \log p(O,Z|\lambda)$. In HMM, the state inference problem is solved by Viterbi algorithm in the spirit of dynamic programming. However, it is difficult to apply this technique to HSM. In Viterbi algorithm, finding the most likely hidden sequence up to time t must depend only on the observed event at t, and the most likely sequences before t. This rule is satisfied in HMM due to its Markov property. But in HSM, we account for the contrastive features between each two observations, and the above rule never holds in HSM.

For this reason, we propose a new technique other than dynamic programming for state inference of HSM. We found that Eq. 8 can be reduced to a quadratic programming problem. Expand $Z = \{z_{i,t}\}$ into an NT-dimensional vector $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_T]$, where $\mathbf{z}_t = [z_{1,t}, z_{2,t}, ..., z_{N,t}]$. Introduce a matrix $D = \{d_{i,t}\}$, where

$$d_{i,t} = \begin{cases} \zeta_{i,t} + \log \pi_i & \text{if } t = 1, \\ \zeta_{i,t} & \text{otherwise.} \end{cases}$$

Similarly, we can expand D into a NT-dimensional vector $\mathbf{d}$. Now, let us consider a tensor $G = \{g_{i,j,t_1,t_2}\}$ where

$$g_{i,j,t_1,t_2} = \begin{cases} \eta_{i,j,t_1,t_2} + \log b_{i,j} & \text{if } t_2 = t_1 + 1, \\ \eta_{i,j,t_1,t_2} & \text{otherwise.} \end{cases}$$

Let $E_{t_1,t_2} = \{g_{:,:,t_1,t_2}\}$ denote a slice of G when t_1, t_2 are fixed. We can unfold G into an $NT \times NT$ matrix $\mathbf{E}$ where,

$$\mathbf{E} = \begin{bmatrix} E_{1,1} & E_{1,2} & \cdots & E_{1,T} \\ E_{2,1} & E_{2,2} & ... & E_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ E_{T,1} & E_{T,2} & \cdots & E_{T,T} \end{bmatrix}$$

Then the maximization of Eq. 8 can be written as the following 0-1 (binary) quadratic programming (QP) problem

$$\max_{\mathbf{z}} f(\mathbf{z}) = \mathbf{z}\mathbf{d}^{\mathrm{T}} + \mathbf{z}\mathbf{E}\mathbf{z}^{\mathrm{T}}, \tag{11}$$
$$\text{subject to: } z_{i,t} \in \{0,1\}, \sum_i z_{i,t} = 1.$$

However, the above 0-1 quadratic programming is still very hard. To circumvent this difficulty, we relax the 0-1 constraint of $\mathbf{z}$, and obtain the following QP problem,

$$\max_{\mathbf{z}} f(\mathbf{z}) = \mathbf{z}\mathbf{d}^{\mathrm{T}} + \mathbf{z}\mathbf{E}\mathbf{z}^{\mathrm{T}}, \tag{12}$$
$$\text{subject to: } 1 \geq z_{i,t} \geq 0, \sum_i z_{i,t} = 1.$$

With the above constraints, Eq. 12 becomes a quadratic programming problem. If matrix $\mathbf{E}$ is negative-definite, this problem can be solved in a polynomial time. We have the following theorem, which discusses the relation between Eq. 11 and Eq. 12, and its proof is given in Appendix.

Theorem 1: The optimal solution of quadratic programming problem of Eq. 12 will be the same as one optimal solution for 0-1 quadratic programming problem of Eq. 11.

D. Probability calculation

Here we study the problem of how to calculate probability $p(O|\lambda)$ of the observed sequence O given model λ, posterior probability $p(q_t = s_i|O,\lambda)$ of t-th observation being state s_i, and posterior probability $p(q_{t_1} = s_i, q_{t_2} = s_j|O,\lambda)$ of joint states. Generally, these probabilities can be calculated as,

$$p(O|\lambda) = \sum_Z p(O,Z|\lambda), \tag{13}$$

$$p(q_t = s_i|O,\lambda) = \sum_{Z(z_{i,t}=1)} p(Z|O,\lambda), \tag{14}$$

$$p(q_{t_1} = s_i, q_{t_2} = s_j|O,\lambda) = \sum_{Z(z_{i,t_1}z_{j,t_2}=1)} p(Z|O,\lambda). \tag{15}$$

To directly calculate the summations of the above equations is very time-costly, since there exist N^T possible Q (Z). In HMM, these problems are solved by forward and backward algorithms. But HSM makes use of contrastive features, which prevent the usage of these fast DP-based algorithms.

In this paper, we consider an approximation method. Let $Z^* = \arg\max_Z p(O,Z|\lambda)$ denote the optimal solution of Eq. 8. Then we can approximate Eq. 13 as

$$p(O|\lambda) \approx \max_Z p(O,Z|\lambda) = p(O,Z^*|\lambda). \tag{16}$$

Introduce variables $r_{i,t}$ and ξ_{i,j,t_1,t_2} to represent the expectations of $z_{i,t}$ and $z_{i,t_1}z_{j,t_2}$ respectively,

$$r_{i,t} = \mathrm{E}[z_{i,t}] = \sum_Z p(Z|O,\lambda) z_{i,t} = p(z_{i,t} = 1|O,\lambda), \tag{17}$$

$$\begin{aligned} \xi_{i,j,t_1,t_2} &= \mathrm{E}[z_{i,t_1}z_{j,t_2}] = \sum_Z p(Z|O,\lambda) z_{i,t_1}z_{j,t_2} \\ &= p(z_{i,t_1}z_{j,t_2} = 1|O,\lambda). \end{aligned} \tag{18}$$

We consider the following 'a winner takes all' approximations of the above variables,

$$r_{i,t} \approx z^*_{i,t}, \tag{19}$$

$$\xi_{i,j,t_1,t_2} \approx z^*_{i,t_1} z^*_{j,t_2}. \tag{20}$$

E. Parameter estimation

In this section, we discuss the problem to estimate the parameters of HSM. Using maximum likelihood estimation, we have

$$\arg\max_\lambda \prod_k p(O^k|\lambda), \tag{21}$$

where O^k denotes the k-th training sequence. There doesn't exist a closed form solution for MLE of HSM. So we adopt EM algorithm [10] for optimization. Note $\{r_{i,t}\}$ and $\{\xi_{i,j,t_1,t_2}\}$ are used as the hidden parameters in EM iteration.

In the E-step, given the old parameters λ^{old}, we need to calculate the distribution of Z denoted by $p(Z|O,\lambda^{\text{old}})$. Since $z_{i,t}$ is binary, this problem is reduced to estimate the expectations $r_{i,t}$ and ξ_{i,j,t_1,t_2}. There are two methods to do

this. One is to estimate the marginal probabilities through summation as in Eq. 14 and Eq. 15. But this is computationally expensive. The other is to use the approximations given by Eq. 19 and Eq. 20. It is noted that these approximations are similar to the Viterbi training [11] of HMM (also known as segmental k-means), where the hidden parameters are determined through Viterbi alignment not by calculating marginal probabilities.

When the hidden parameters $r^k_{i,t}$ and ξ^k_{i,t_1,j,t_2} are given, we can find the model parameters through maximizing the auxiliary function $Q(\lambda, \lambda^{\text{old}})$,

$$Q(\lambda, \lambda^{\text{old}}) = \sum_k \sum_Z p(Z|O^k, \lambda^{\text{old}}) \log p(Z, O^k|\lambda) \tag{22}$$

$$= \sum_k \{ \sum_{i=1}^{N} r^k_{i,1} \log \pi_i + \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \log b_{i,j} \xi^k_{i,j,t,t-1} + \sum_{t=1}^{T} \sum_{i=1}^{N} \zeta^k_{i,t} r^k_{i,t} + \sum_{1 \le t_1, t_2 \le T} \sum_{i=1}^{N} \sum_{j=1}^{N} \eta^k_{i,j,t_1,t_2} \xi^k_{i,j,t_1,t_2} \}. \tag{23}$$

Then the optimal parameters can be calculated by,

$$\pi_i = \frac{\sum_k r^k_{i,1}}{\sum_k \sum_{j=1}^{N} r^k_{j,1}}, \tag{24}$$

$$b_{i,j} = \frac{\sum_k \sum_{t=2}^{T} \xi^k_{i,j,t-1,t}}{\sum_k \sum_{m=1}^{N} \sum_{t=2}^{T} \xi^k_{m,j,t-1,t}}, \tag{25}$$

$$\mu^a_i = \frac{\sum_k \sum_{t=1}^{T} \bar{o}^k_t r^k_{i,t}}{\sum_k \sum_{t=1}^{T} r^k_{i,t}}, \tag{26}$$

$$\Sigma^a_i = \frac{\sum_k \sum_{t=1}^{T} r^k_{i,t} (\bar{o}^k_t - \mu^a_i)(\bar{o}^k_t - \mu^a_i)^{\mathrm{T}}}{\sum_k \sum_{t=1}^{T} r^k_{i,t}}, \tag{27}$$

$$\mu^c_{i,j} = \frac{\sum_k \sum_{t_1,t_2} c^k_{t_1,t_2} \xi^k_{i,j,t_1,t_2}}{\sum_k \sum_{t_1,t_2} \xi^k_{i,j,t_1,t_2}}, \tag{28}$$

$$\Sigma^c_i = \frac{\sum_k \sum_{t_1,t_2} (c^k_{t_1,t_2} - \mu^c_{i,j})(c^k_{t_1,t_2} - \mu^c_{i,j})^{\mathrm{T}} \xi^k_{i,j,t_1,t_2}}{\sum_k \sum_{t_1,t_2} \xi^k_{i,j,t_1,t_2}}. \tag{29}$$

IV. Experiments

We carry out two preliminary experiments to examine HSM on labeling sequences. It is noted that the previous structural representation cannot conduct such tasks.

A. Experiment 1 with generated and transformed sequences

The first experiment examines the performance of HSM with artificially generated and transformed sequences. As preparation, we calculate the Gaussian distributions of cepstrum features for six symbols, i.e., five Japanese vowels ('a','e','i','o','u') and silence ('sl'). Using the six symbols, we randomly generate a set of strings, each of which contains 16 symbols. Then the corresponding cepstrum features of these strings are obtained by using the Gaussian models to generate five frame vectors for each symbol. After that, we perform acoustic transformation on the generated cepstrum features as if the features are generated by different speakers. The acoustic transformation is realized by frequency warping, which corresponds to multiplication of cepstrum vectors by a specific type of matrix [8]. The elements of the matrix are functions of warping parameter α [8] and, by changing the value of α, we can lengthen/shorten the vocal tract length of speech samples. Here, the value of α in [-0.5,0.5] is randomly selected from a uniform distribution. In this way, we generate a set of strings which are acoustically realized by different speakers. Using this procedure, a set of transformed cepstrum sequences are prepared for training and another set for testing. It should be noted that these sequences are different strings and are acoustically realized by different speakers.

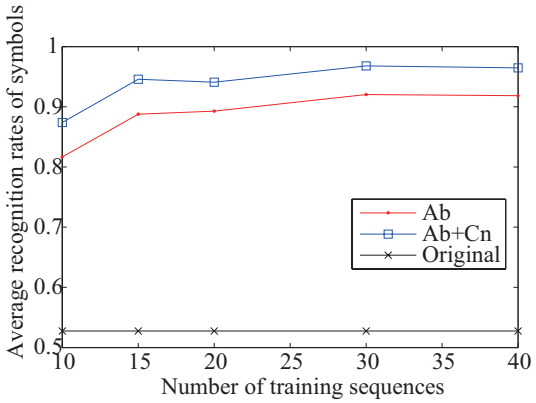

Fig. 4. Average symbol-based recognition rates for randomly generated and transformed sequences.

We train a single six-state HSM from the sequences in the training set. It is noted that since label (symbol) and boundary information of every sequence is known, we can use Eq. 26, Eq. 27, Eq. 28, and Eq. 29 to directly estimate the distribution parameters of the absolute and contractive features without EM iterations. Once the HSM is trained, we use the QP-based state inference method proposed in Section III-C to estimate the symbol (state) information of every testing cepstrum sequence. In other words, each input sequence is aligned to the HSM. The testing set contains 20 sequences. We change the number of training sequences from 10 to 40. For each case, we repeat the experiments 20 times. The average symbol-based recognition rates are shown in Fig. 4, where 'Ab' represents the use of absolute features only, and 'Ab+Cn' the use of both absolute and contrastive features. 'Original' means using the original Gaussian distributions for training. As discussed in the beginning of Section III, the absolute feature only is essentially the same as HMM. As one can see, the combination of both absolute and contrastive features has the best performance.

B. Experiment 2 with Japanese vowel utterances

In the second experiment, we examine the recognition performances with a database of continuously connected Japanese

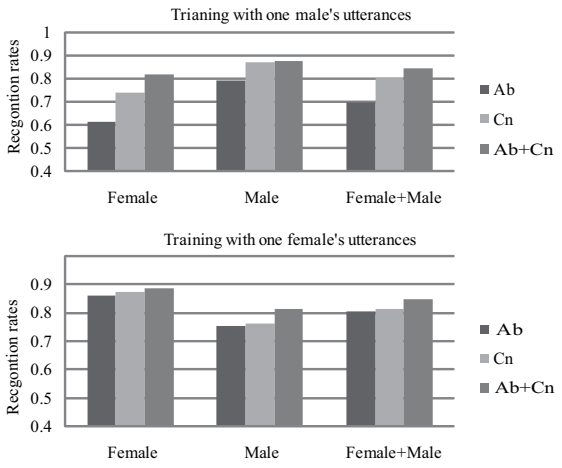

Fig. 5. State recognition rates for connected Japanese vowel utterances.

vowel utterances. It is known that acoustic features of vowel sounds exhibit larger between-speaker variations than consonant sounds. Each word in the data set is a concatenation of the five Japanese vowels 'a','e','i','o' and 'u', such as 'aeiou','uoaie', etc. So there are totally 120 words. The utterances of 16 speakers (8 males and 8 females) were recorded. Every speaker provides 5 utterances for each word. The total number of utterances is 16×120×5 =9,600. For each utterance, we calculate twelve Mel-cepstrum features and one power coefficient. Then ML-based decomposition is used to convert cepstrum vectors into a sequence of 25 Gaussian distributions (sub-segments) [4], [3]. We label each distribution as 'a','e','i','o', 'u' or 'sl' by forced alignment with speaker-dependent phoneme HMMs.

We train independently a 6-state HSM with 600 (120×5) utterances of a male speaker and another HSM for a female speaker. For each HSM, we examine its state recognition rates for utterances of other male speakers, other female speakers, and both. The results are summarized in Fig. 5. Like the results of experiment 1, we found that 'Ab+Cn' achieves the best recognition rates. Moreover, the contrastive features have better performance than the absolute features.

V. Conclusions

This paper proposes Hidden Structure Model (HSM) for sequence data. HSM generalizes our previous structural representation into a probabilistic framework, which accounts for both absolute and contrastive features. Like HMM, HSM makes use of hidden states. Different from HMM, HSM contains the distributions of contrastive features. We also develop algorithms for state inference, probability calculation, and parameter estimation of HSM. Due to the usage of contrastive features, we cannot use dynamic programming to develop HMM-like algorithms, such as Viterbi algorithm, forward and backward algorithm, and Baum-Welch algorithm. In this paper, we formulate the state inference into a quadratic programming problem, and develop approximation methods for probability calculation and parameter estimation. We conducted two preliminary experiments to examine the performance of HSMs. One is based on artificially generated sequences, the other makes use of the connected Japanese vowel utterances. The results show the usefulness of HSM and the advantages of combining absolute and contrastive features. The results of this paper are preliminarily and limited. In the future, we are going to improve the model and algorithms of HSM, and examine HSM with larger database.

Appendix

Proof: Let $\mathbf{z}'$ denote the optimal solution for Eq. 11 and $\mathbf{z}^*$ denote the optimal solution for Eq. 12. Since the constraints of Eq. 12 are more general than Eq. 11, we have,

$$f(\mathbf{z}^*) \geq f(\mathbf{z}'). \tag{30}$$

Now consider a distribution $p^*(\mathbf{z})$ of discrete variables $\mathbf{z}$ with $p^*(z_{i,t} = 1) = z^*_{i,t}$. Calculate the expectation of $f(\mathbf{z})$ under $p^*(\mathbf{z})$ as,

$$E_{p^*(\mathbf{z})}[f(\mathbf{z})] = E_{p^*(\mathbf{z})}[\mathbf{z}\mathbf{d}^{\mathrm{T}} + \mathbf{z}\mathbf{E}\mathbf{z}^{\mathrm{T}}] = f(\mathbf{z}^*). \tag{31}$$

On the other hand,

$$E_{p^*(\mathbf{z})}[f(\mathbf{z})] = \sum_{\mathbf{z}} p^*(\mathbf{z}) f(\mathbf{z}) = f(\mathbf{z}^*). \tag{32}$$

Since $0 \leq p^*(\mathbf{z}) \leq 1$ for all z, there must exist one $\mathbf{z}''$ such that $f(\mathbf{z}'') \geq f(\mathbf{z}^*)$. Thus $f(\mathbf{z}') \geq f(\mathbf{z}'') \geq f(\mathbf{z}^*)$. Recall Eq. 30, $f(\mathbf{z}') = f(\mathbf{z}^*)$ must hold. Therefore, the optimal solutions for Eq. 11 and Eq. 12 must be the same. ■

References

[1] N. Minematsu, "Mathematical Evidence of the Acoustic Universal Structure in Speech," *Proc. ICASSP*, pp. 889–892, 2005.

[2] Y. Qiao and N. Minematsu, "f-divergence is a generalized invariant measure between distributions," *Proc. INTERSPEECH*, pp. 1349–1352, 2008.

[3] Y. Qiao, S. Asakawa, and N. Minematsu, "Random Discriminant Structure Analysis for Automatic Recognition of Connected Vowels," *Proc. ASRU*, pp. 576–581, 2007.

[4] S. Asakawa, N. Minematsu, and K. Hirose, "Multi-stream parameterization for structural speech recognition," *Proc. ICASSP*, pp. 4097–4100, 2008.

[5] S. Saito, D. Asakawa, N. Minematsu, and K. Hirose, "Structure to speech – speech generation based on infant-like vocal imitation," *Proc. INTERSPEECH*, pp. 1837–1840, 2008.

[6] N. Minematsu, S. Asakawa, and K. Hirose, "Structural representation of the pronunciation and its use for CALL," *Proc. of IEEE Spoken Language Technology Workshop*, pp. 126–129, 2006.

[7] I. Csiszar, "Information-type measures of difference of probability distributions and indirect," *Stud. Sci. Math. Hung.*, vol. 2, pp. 299–318, 1967.

[8] M. Pitz and H. Ney, "Vocal Tract Normalization Equals Linear Transformation in Cepstral Space," *IEEE Trans. SAP*, vol. 13, no. 5, pp. 930–944, 2005.

[9] Y. Qiao, N. Shimomura, and N. Minematsu, "Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons," *Porc. ICASSP*, pp. 3989–3992, 2008.

[10] A. Dempster, N. Laird, D. Rubin *et al.*, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[11] L. Rabiner, J. Wilpon, and B. Juang, "A segmental K-means training procedure for connected word recognition," *AT & T technical journal*, vol. 65, no. 3, pp. 21–31, 1986.