Vol.2009-SLP-77 No.5
2009/7/17

An Investigation of Hidden Structure Model

Yu QIAO ,[†1] Masayuki SUZUKI [†2]
and Nobuaki MINEMATSU[†1]

In recent years, we have been working toward a structural representation of speech using contrastive features that are robust to non-linguistic variations. This paper generalizes the structural representation to Hidden Structure Model (HSM) by introducing hidden states and probabilistic calculations. HSM not only can solve miss-alignment problems of events, but also can conduct structure-based decoding, which allows us to apply HSM to general speech recognition tasks. This paper focuses on the fundamental theories of HSM. Different from HMM, HSM accounts for both the absolute and contrastive aspects of an input sequence. We show that the state inference of HSM can be formulated as a quadratical programming problem. We also introduce EM algorithm to estimate the parameters of HSM.

1. Introduction

Speech signals inevitably exhibit large non-linguistic variations, caused by the difference of speakers, communication channels, environment noise etc. This poses one of the major challenging problems in speech engineering. To deal with these variations, modern speech recognition approaches mainly make use of statistical methods (such as GMM, 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 data. But it is well-known that the performance of speech recognizers will drop significantly if there exists a mismatch. Contrary to this is 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 learn speech with very biased training data from mothers and fathers. This is also partly supported by recent advances in neurosciences, which show that the linguistic aspect of speech and the non-linguistic aspect are processed separately in auditory cortex[14].

Inspired by these facts, Minematsu proposed an invariant structural representation of speech signals which aims at removing the non-linguistic factors in speech signals[6]. Different from classical speech models, the structural representations make use of contrastive features (f-divergence) to model the global and dynamic aspects of speech and discard the local and static features. It can be proved that these contrastive features are invariant to transformations and thus are robust to non-linguistic variations. We have already demonstrated the effectiveness of this representation in ASR[2),9)], speech synthesis[13] CALL[7], and dialect analysis[5].

In this paper, we generalize the structural representation into Hidden Structure Model (HSM) by introducing hidden states and probabilistic analysis. This generalization allows us to overcome two limitations of previous structural representations. Compared with these, HSM unifies structure construction and structure comparison into a single framework, and avoids the misalignment of events. Moreover, the introduction of hidden states enables HSM to conduct structure-based decoding, different from previous structure-based matching. 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 absoulte and contrastive features. This paper proposes the probabilistic formulation of HSM and develops the algorithms for state inference, probability calculation and parameter estimation of HSM.

2. Review of previous structural representations

In this section, we give a brief overview on the invariant structure theory and how to calculate structural representations from utterances[6].

2.1 Theory of invariant structure

Consider feature space X and pattern P in X. Suppose P is composed of a sequence of K events $\{p_i\}_{i=1}^{K}$. Each event is described as a distribution $p_i(x)$

[†1] Graduate School of Information Science and Technology, The University of Tokyo
[†2] Graduate School of Engineering, The University of Tokyo

in the feature space. Note x can have multiple dimensions. Assume there is an invertible transformation $h : X \rightarrow Y$ (linear or nonlinear) which maps x into y. In this way, pattern P in X is transformed to pattern Q in Y, and event $p_i(x)$ is converted to event $q_i(y)$. Thus if we can find invariant metrics in both space X and space Y, these metrics can yield robust features for classification.

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[3] 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: $D_f(q_i, q_j) = D_f(p_i, p_j)$[10]. Moreover, we found that all the invariant integration measures $\oint M(p_i, p_j)dx$ must be in the form of f-divergence[10]. From pattern P, we can obtain a $K \times K$ divergence matrix $\mathcal{D}^P$ with $\mathcal{D}^P(i, j) = D_f(p_i, p_j)$ and $\mathcal{D}^P(i, i) = 0$. Then $\mathcal{D}^P$ provides a structural representation of pattern P. Similarly, we can obtain structure representation $\mathcal{D}^Q$ for pattern Q. Then we have that $\mathcal{D}^Q \equiv \mathcal{D}^P$, which indicates that the structural representation is invariant to transformations.

2.2 Structuralization of an utterance

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 HMM is regarded as event p_i. Thirdly we calculate f-divergences between each event pair. These divergences will form a symmetric 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. It is easy to see that this structural representation must be invariant to transformations in feature space. In speech engineering, the non-linguistic speech variations are also modeled as transformation of cepstrum feature space. Microphones and environment distortion modifies the cepstrum feature with an additive vector. And vocal tract length difference is often modeled as linear transformation of 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,

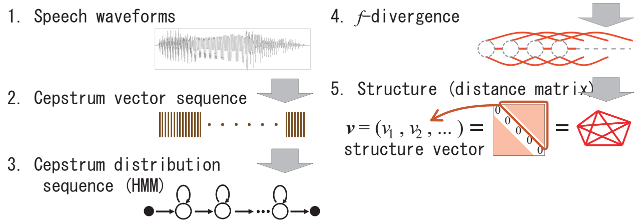
1. Speech waveforms
2. Cepstrum vector sequence
3. Cepstrum distribution sequence (HMM)
4. f-divergence
5. Structure (distance matrix)
v = (v1 , v2 , ...) =
structure vector
0
0
0
0
0
=

Fig. 1 Framework of structure construction.

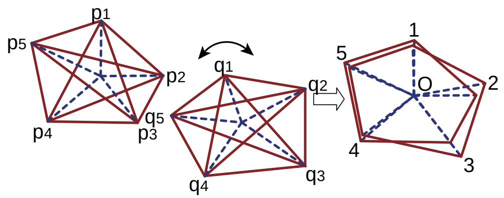

Fig. 2 Structure (utterance) matching by shift and rotation.

$$D(P, Q) = \sum_{i,j} |\mathcal{D}^P(i, j) - \mathcal{D}^Q(i, j)|^2. \qquad (2)$$

It can be shown that the acoustic matching score of two utterances after shift and rotation can be approximated as the difference of the two structure vectors (Fig.2)[6]. It is noted that, different from speaker adaptive training (SAT), structure matching doesn't need to explicitly estimate transformation parameters for model adaptation or feature normalization.

3. Hidden Structure Model

In the previous structural representation, the distribution sequences are calculated for each utterance independently. 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 the phoneme 'i', but q_3 of Q may come from the phoneme 'u'. Another limitation of structural representation is that its doesn't include any label or category information of each event. Although the word recognition problem can be reduced to a structure matching, it is difficult to generalize this technique for general speech recognition tasks.

We notice that HMM doesn't have the above limitations. HMM avoids the misalignment problem by using DP-matching to align a speech stream with a sequence of HMM distributions. Moreover, HMM has flexible Viterbi decoding to estimate the most probable hidden states, which makes HMM suitable for solving general phoneme recognition tasks. Remind that the main advantage of structural representation is that it makes use of contrastive features, which are robust to non-linguist variations. 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 hidden states of observations and take account for the labels of these hidden states. Unlike HMM, HSM models the distributions of absolute and contrastive features, which makes it more robust to speaker differences.

3.1 Preprocessing of speech sequences for HSM

The contrastive features have to be calculated from 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 $O = o_1, o_2, ..., o_T$ in a preprocessing step (Fig. 3). Generally, we can use agglomerative clustering algorithm (ACA)[11] or HMM-based decomposition for sequence division[1),9)]. 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 2nd method, each segment is modeled as a Gaussian distribution $N(\bar{o}_t, V_t)$. For every two segments o_{t_1} and o_{t_2}, we use c_{t_1,t_2} to denote the contrastive feature between them.

3.2 Introduction of Hidden Structure Model

Generally speaking, HSM is a probabilistic model for sequence data, which takes account for the distributions of both absolute and contrastive features. To begin with, we formally describe 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 denote the state corresponding to 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 $b_{i,j} = p(q_{t+1} = s_j | q_t = s_i)$ $(1 \leq i, j \leq N)$.

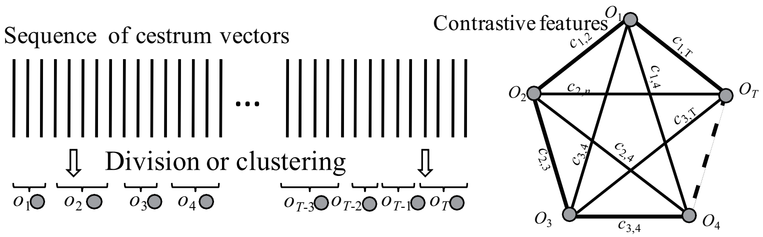

Fig. 3 Preprocessing of cepstrum sequence.

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

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,

$$p(\bar{o}_t | q_t = s_j) = N(\bar{o}_t | \mu_j^a, \Sigma_j^a). \tag{3}$$

Let $A = \{\mu_j^a, \Sigma_j^a\}$ denote the set of AOP parameters.

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.[2),10)]) between o_{t_1} and o_{t_2}. COP is assumed to have a Gaussian form,

$$p(c_{t_1,t_2} | q_{t_1} = s_i, q_{t_2} = s_j) = N(c_{t_1,t_2} | \mu_{i,j}^c, \Sigma_{i,j}^c). \tag{4}$$

Let $C = \{\mu_{i,j}^c, \Sigma_{i,j}^c\}$ denote the set of COP parameters.

One can see that items 1)-4) are the same as classical HMM, but item 5) is used to describe the 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 conditional probability of O given model λ and state sequence Q as,

$$p(O | Q, \lambda) = \underbrace{\prod_{t=1}^{T} p(\bar{o}_t | q_t)}_{\text{Absolute part}} \underbrace{\prod_{1 \leq t_1, t_2 \leq T} p(c_{t_1,t_2} | q_{t_1}, q_{t_2})}_{\text{Contrastive part}}. \tag{5}$$

An example of HSM is depicted in Fig. 4. Note if we remove the contrastive part of Eq. 5, this probability calculation will be the same as that of HMM. On the other hand, if we remove the absolute part, Eq. 5 reduces to a pure model

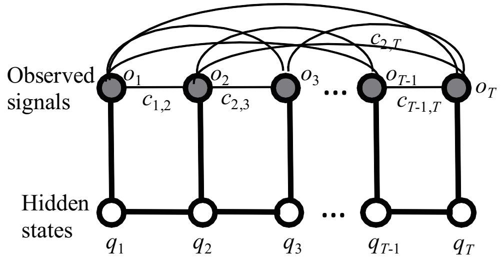

Fig. 4 Diagram of HSM with five states. (HMM contains only the thick lines.)

of structural representation.

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. 5 into

$$p(O|Q,\lambda) = p(O|Z,\lambda) = \prod_{t=1}^{T}\prod_{i=1}^{N} p(\bar{o}_t|s_i)^{z_{i,t}} \prod_{1\le t_1,t_2\le T}\prod_{i=1}^{N}\prod_{j=1}^{N} p(c_{t_1,t_2}|s_i,s_j)^{z_{i,t_1}z_{j,t_2}}. \tag{6}$$

We can calculate the probability of state sequence like HMM[⋆1]

$$p(Q|\lambda) = p(Z|\lambda) = p(q_1)\prod_{t=2}^{T} p(q_t|q_{t-1}) = \prod_{i=1}^{N} p(s_i)^{z_{i,1}} \prod_{t=2}^{T}\prod_{i=1}^{N}\prod_{j=1}^{N} p(s_i|s_j)^{z_{i,t}z_{j,t-1}}. \tag{7}$$

Therefore, we have

$$\begin{aligned} p(O,Q|\lambda) &= p(O,Z|\lambda) = p(O|Z,\lambda)p(Z|\lambda) \\ &= \prod_{i=1}^{N} p(s_i)^{z_{i,1}} \prod_{t=1}^{T}\prod_{i=1}^{N} p(\bar{o}_t|s_i)^{z_{i,t}} \prod_{t=2}^{T}\prod_{i=1}^{N}\prod_{j=1}^{N} p(s_i|s_j)^{z_{i,t}z_{j,t-1}} \\ & \prod_{1\le t_1,t_2\le T}\prod_{i=1}^{N}\prod_{j=1}^{N} p(c_{t_1,t_2}|s_i,s_j)^{z_{i,t_1}z_{j,t_2}}. \end{aligned} \tag{8}$$

Calculate the log of the above equation,

$$\begin{aligned} \log p(O,Z|\lambda) = \sum_{i=1}^{N} z_{i,1}\log\pi_i + \sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} z_{i,t}z_{j,t-1}\log b_{i,j} + \\ \sum_{t=1}^{T}\sum_{i=1}^{N}\zeta_{i,t}z_{i,t} + \sum_{1\le t_1,t_2\le T}\sum_{i=1}^{N}\sum_{j=1}^{N}\eta_{i,j,t_1,t_2}z_{i,t_1}z_{j,t_2}. \end{aligned} \tag{9}$$

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 algorithms to solve the three problems of HSM, namely, state inference, probability calculation and parameter estimation.

3.3 State inference

Given model λ and observed stream O, the objective of state inference is to determine Z which maximizes the following conditional probabilty,

$$\arg\max_{Z} p(Z|O,\lambda). \tag{10}$$

Using Bayesian theory, we have

$$p(Z|O,\lambda) = \frac{p(O,Z|\lambda)}{p(O|\lambda)} \propto p(O,Z|\lambda). \tag{11}$$

Thus the problem can be reduced to find Z which maximizes Eq. 9, $\max_Z \log p(O,Z|\lambda)$. 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 for HSM. In Viterbi algorithm, finding the most likely hidden sequence up to time point 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. The above rule is never held in HSM.

For this reason, we propose a new technique other than dynamic programming for HSM. We found that Eq. 9 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

[⋆1] More generally, we can take account for the transmission probabilities of every two time points and define $p(Z|\lambda) = \prod_{t_1,t_2=1}^{T}\prod_{i,j=1}^{N} h(q_{t_1} = s_i|q_{t_2} = s_j)^{z_{i,t_1}z_{j,t_2}}$, where h denotes a certain cost function. This general definition (of state probability) can be applied to the following analysis.

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. 9 can be written as the following 0-1 (binary) quadratic programming (QP) problem

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

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

$$\max_{\mathbf{z}} f(\mathbf{z}) = \mathbf{z}\mathbf{d}^{\mathrm{T}} + \mathbf{z}\mathbf{E}\mathbf{z}^{\mathrm{T}}, \qquad (13)$$
$$\text{subject to: } 1 \geq z_{i,t} \geq 0, \sum_i z_{i,t} = 1.$$

With the above constraints, Eq. 13 becomes a quadratic programming problem. If matrix $\mathbf{E}$ is negative-definite, this problem can be solved in a polynomial time. It can be shown that the relaxed QP of Eq. 13 will always have the same optimal solution as 0-1 QP of Eq. 12. The proof is omitted due to space limitation.

3.4 Probability calculation

In this section, 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.

Using marginal probability, we have

$$p(O|\lambda) = \sum_Z p(O,Z|\lambda), \qquad (14)$$

$$p(q_t = s_i|O,\lambda) = p(z_{i,t} = 1|O,\lambda) = \sum_{Z(z_{i,t}=1)} p(Z|O,\lambda), \qquad (15)$$

$$p(q_{t_1} = s_i, q_{t_2} = s_j|O,\lambda) = p(z_{i,t_1}z_{j,t_2} = 1|O,\lambda) = \sum_{Z(z_{i,t_1}z_{j,t_2}=1)} p(Z|O,\lambda). \qquad (16)$$

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

In this paper, we consider an approximation method. Let Z^* denote the optimal solution of Eq. 9, i.e., $\arg\max_Z p(O,Z|\lambda)$. Then we can approximate Eq. 14 as

$$p(O|\lambda) \approx \max_Z p(O,Z|\lambda) = p(O,Z^*|\lambda). \qquad (17)$$

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,

$$r_{i,t} = \mathrm{E}[z_{i,t}] = \sum_Z p(Z|O,\lambda)z_{i,t} = p(z_{i,t} = 1|O,\lambda), \qquad (18)$$

$$\xi_{i,j,t_1,t_2} = \mathrm{E}[z_{i,t_1}z_{j,t_2}] = \sum_Z p(Z|O,\lambda)z_{i,t_1}z_{j,t_2} = p(z_{i,t_1}z_{j,t_2} = 1|O,\lambda). \qquad (19)$$

With these, we consider the following a 'winner takes all' approximations,

$$r_{i,t} \approx z^*_{i,t}, \qquad (20)$$
$$\xi_{i,j,t_1,t_2} \approx z^*_{i,t_1}z^*_{j,t_2}. \qquad (21)$$

3.5 Parameter estimation

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

$$\arg\max_\lambda \prod_k p(O^k|\lambda), \qquad (22)$$

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[4] for optimization. Note $\{r_{i,t}\}$ and $\{\xi_{i,j,t_1,t_2}\}$ are the hidden parameters in EM iteration here.

In the E-step, given the old parameters λ^{old}, we need to calculate the distribution of Z denoted by $p(Z|O,\lambda^{\mathrm{old}})$. 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. 15 and Eq. 16. The other is to use the approximations given by Eq. 20 and Eq. 21. It is noted that these approximations are similar to the Viterbi training[12] 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 are given, we can find model parameters which maximizes the auxiliary function $Q(\lambda, \lambda^{\text{old}})$,

$$Q(\lambda, \lambda^{\text{old}}) = \sum_k \sum_Z p(Z|O^k, \lambda^{\text{old}}) \log p(Z, O^k|\lambda). \tag{23}$$

With hidden parameters $r^k_{i,t}$ and ξ^k_{i,t_1,j,t_2} for O^k, we have

$$\begin{aligned} Q(\lambda, \lambda^{\text{old}}) = \sum_k \{ \sum_{i=1}^{N} r^k_{i,1} \log \pi_i + \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \log b_{i,j} \xi^k_{i,j,t,t-1} + \\ \sum_{t=1}^{T} \sum_{i=1}^{N} \zeta^k_{i,t} r^k_{i,t} + \sum_{1 \le t_1, t_2 \le T} \sum_{i=1}^{N} \sum_{j=1}^{N} \eta^k_{i,j,t_1,t_2} \xi^k_{i,j,t_1,t_2} \}. \end{aligned} \tag{24}$$

Then the optimal parameters can be calculated by,

$$\pi_i = \frac{\sum_k r^k_{i,1}}{\sum_k \sum_{j=1}^{N} r^k_{j,1}}, \tag{25}$$

$$b_{i,j} = \frac{\sum_k \sum_{t=2}^{T} \xi^k_{i,j,t-1,t}}{\sum_k \sum_{m=1}^{N} \sum_{t=2}^{T} \xi^k_{m,j,t-1,t}}, \tag{26}$$

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

$$\Sigma^a_i = \frac{\sum_k \sum_{t=1}^{T} r^k_{i,t} (\bar{o}^k_t - \mu^a_i)(\bar{o}^k_t - \mu^a_i)^{\mathrm{T}}}{\sum_k \sum_{t=1}^{T} r^k_{i,t}}, \tag{28}$$

$$\mu^c_{i,j} = \frac{\sum_k \sum_{t_1,t_2} c^k_{t_1,t_2} \xi^k_{i,j,t_1,t_2}}{\sum_k \sum_{t_1,t_2} \xi^k_{i,j,t_1,t_2}}, \tag{29}$$

$$\Sigma^c_i = \frac{\sum_k \sum_{t_1,t_2} (c^k_{t_1,t_2} - \mu^c_{i,j})(c^k_{t_1,t_2} - \mu^c_{i,j})^{\mathrm{T}} \xi^k_{i,j,t_1,t_2}}{\sum_k \sum_{t_1,t_2} \xi^k_{i,j,t_1,t_2}}. \tag{30}$$

4. 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 parameters 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 algorithms for probability calculation and parameter estimation. This paper only focuses on the fundamental theories of HSM. In the next, we are going to examine the proposed model and algorithms through experiments.

References

1) Asakawa, S.: A study on word speech recognition based on structural representation of speech, PhD Thesis, The Univ. of Tokyo (2008, to appear).

2) Asakawa, S., Minematsu, N. and Hirose, K.: Multi-stream parameterization for structural speech recognition, *Proc. ICASSP*, pp.4097–4100 (2008).

3) Csiszar, I.: Information-type measures of difference of probability distributions and indirect, *Stud. Sci. Math. Hung.*, Vol.2, pp.299–318 (1967).

4) Dempster, A., Laird, N., Rubin, D. 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).

5) Ma, X., Nemoto, A., Minematsu, N., Qiao, Y. and Hirose, K.: Structural analysis of dialects, sub-dialects, and sub-sub-dialects of Chinese, *INTERSPEECH* (2009).

6) Minematsu, N.: Mathematical Evidence of the Acoustic Universal Structure in Speech, *Proc. ICASSP*, pp.889–892 (2005).

7) Minematsu, N., Asakawa, S. and Hirose, K.: Structural representation of the pronunciation and its use for CALL, *Proc. of IEEE Spoken Language Technology Workshop*, pp.126–129 (2006).

8) Pitz, M. and Ney, H.: Vocal Tract Normalization Equals Linear Transformation in Cepstral Space, *IEEE Trans. SAP*, Vol.13, No.5, pp.930–944 (2005).

9) Qiao, Y., Asakawa, S. and Minematsu, N.: Random discriminant structure analysis for automatic recognition of connected vowels, *Proc. of ASRU*, pp.576–581 (2007).

10) Qiao, Y. and Minematsu, N.: *f*-divergence is a generalized invariant measure between distributions, *Proc. INTERSPEECH* (2008).

11) Qiao, Y., Shimomura, N. and Minematsu, N.: Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons, *ICASSP*, pp.3989–3992 (2008).

12) Rabiner, L., Wilpon, J. and Juang, B.: A segmental K-means training procedure for connected word recognition, *AT & T technical journal*, Vol.65, No.3, pp.21–31 (1986).

13) Saito, D.Asakawa, S., Minematsu, N. and Hirose, K.: Structure to speech – speech generation based on infant-like vocal imitation, *Proc. INTERSPEECH*, pp.1837–1840.

14) Scott, S.K. and Johnsrude, I.S.: The neuroanatomical and functional organization of speech perception, *Trends in Neurosciences*, Vol.26, No.2, pp.100–107 (2003).