Parallel Inference of Dirichlet Process Gaussian Mixture Models for Unsupervised Acoustic Modeling: A Feasibility Study

Hongjie Chen[1], *Cheung-Chi Leung*[2], *Lei Xie*[1], *Bin Ma*[2], *Haizhou Li*[2]

[1]Shaanxi Provincial Key Laboratory of Speech and Image Information Processing, School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Institute for Infocomm Research, A*STAR, Singapore
{hjchen,lxie}@nwpu-aslp.org, {ccleung,mabin,hli}@i2r.a-star.edu.sg

Abstract

We adopt a Dirichlet process Gaussian mixture model (DPGMM) for unsupervised acoustic modeling and represent speech frames with Gaussian posteriorgrams. The model performs unsupervised clustering on untranscribed data, and each Gaussian component can be considered as a cluster of sounds from various speakers. The model infers its model complexity (i.e. the number of Gaussian components) from the data. For computation efficiency, we use a parallel sampler for the model inference. Our experiments are conducted on the corpus provided by the zero resource speech challenge. Experimental results show that the unsupervised DPGMM posteriorgrams obviously outperform MFCC, and perform comparably to the posteriorgrams derived from language-mismatched phoneme recognizers in terms of the error rate of ABX discrimination test. The error rates can be further reduced by the fusion of these two kinds of posteriorgrams.

Index Terms: Bayesian nonparametrics, Gibbs sampling, acoustic unit discovery, Gaussian posteriorgrams, ABX discrimination

1. Introduction

In many state-of-the-art speech applications, a considerable amount of labeled speech data and language-specific linguistic knowledge (such as phoneme definition and pronunciation dictionary) are needed to build reliable statistical models. It is time-consuming and expensive to acquire these resources. Even worse, in some languages, there is no written form and the linguistic knowledge may be even completely absent. This leads to the increasing interest in unsupervised speech processing in recent years. Acoustic pattern matching [1–3] and unsupervised discovery of subword units [4–13] from the raw speech of a low-resource language are being studied. It is generally assumed that only untranscribed data is available for the target language in the study. These techniques have been applied to applications, such as topic segmentation [14, 15], spoken term detection [4, 16], spoken document classification [17] and summarization [18], etc.

Acoustic pattern matching was first studied with spectral features (e.g. MFCC) [1]. Later on, GMM posteriorgrams, one kind of model-based posteriorgrams, was introduced to acoustic pattern discovery [2, 16]. This kind of posterior features has been shown less sensitive to speaker variation and better performance than spectral features. It is suitable for the case when only untranscribed data is available. Noteworthily the posteriorgrams derived from phoneme recognizers and unsupervised subword models have also been used in low-resource applications [7, 8, 19]. In additional to posteriorgrams, there are works on frame-based embedding representation motivated by manifold learning [20–22] and deep learning [10, 11, 23].

In this paper, we are interested in deriving posterior features whose model adapts to the untranscribed data. GMM posteriorgram is a suitable choice where each Gaussian component can be considered as a cluster of sounds from various speakers [2]. However, in real situation, development data may not be available, so the model complexity (i.e. the number of Gaussian components) cannot be known easily. This motivates us to derive posteriorgrams from Dirichlet process Gaussian mixture models (DPGMMs). DPGMM is a mixture model with infinite components. It has been successfully applied to unsupervised lexical clustering of speech segments [24]. We expect that DPGMM can also serve well in frame-level clustering so that it can provide effective features that highlight the linguistic content with their good speaker-independence in speech pattern discovery.

Moreover, we adopt a parallel sampler [25] for the DPGMM inference in our study. The training of DPGMM is unavoidably slow because of the sampling based inference. Moreover, a Bayesian nonparametric model relies on the amount of training data to fit a suitable model. Thus an efficient inference algorithm which is scalable to a large amount of training data is desired. Note that in addition to the use of DPGMMs for lexical clustering, a Bayesian nonparametric model [19] that jointly performs segmentation, subword unit discovery and modeling of the subword units for untranscribed speech has been proposed. However, parallel inference of this kind of models has never been considered in these speech applications.

We evaluate our proposed features on a minimal-pair ABX phoneme discrimination task [26, 27]. This task, which only requires the generated features and a proper distance metric for the features, provides a straightforward way to measure the discriminability between two sound categories. In some previous studies, the learned subword models are evaluated by their clustering performance (e.g. measured by purity) with reference manual subword labels. In this case, there is an assumption on language-specific knowledge (e.g. number of subword units) in the generated features and the evaluation metric. This evaluation approach is not suitable for evaluating the features derived from a Bayesian nonparametric model. Alternatively, many proposed features [8, 19] derived from unsupervised subword modeling are usually evaluated with a relevant application, such as spoken term detection. However, some postprocessing techniques (e.g. score normalization and pseudo-relevance feedback in spoken term detection), which are usually application-specific, may be able to tolerate the defects in the features.

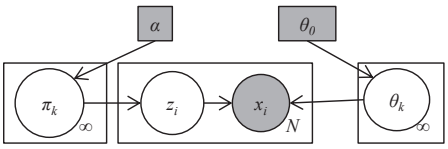

Figure 1: *Graphical representation of Dirichlet process Gaussian Mixture Model (DPGMM).*

As a result, the evaluation metrics of the application, such as mean average precision (MAP) and actual term-weighted value (ATWV) in spoken term detection, may not directly indicate the effectiveness of the proposed features.

2. Dirichlet Process Gaussian Mixture Model

We employ Dirichlet process Gaussian mixture model (DPGMM), also referred to as infinite Gaussian mixture model (IGMM), to conduct frame-level clustering and extract posteriorgrams. DPGMM is a Bayesian nonparametric model which can automatically learn the number of components according to the observed data. It is more suitable for an unsupervised scenario in which no language-specific knowledge exists, or there is no development data in the target language.

2.1. Definition of Generative Process

The graphical representation of DPGMM is illustrated in Figure 1. Given a group of observations, $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, a DPGMM is constructed according to the following generative process of $\mathcal{X}$:

(1) Generate the mixing weights $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^\infty$ according to a stick-breaking process [28];

(2) Generate a set of parameters $\{\boldsymbol{\theta}_k\}_{k=1}^\infty$ of a Gaussian mixture model (GMM) according to their prior distribution named Normal-inverse-Wishart (NIW) distribution [29] with parameters $\boldsymbol{\theta}_0$;

(3) For each observation $\mathbf{x}_i$ to be generated, assign a component label z_i according to the mixing proportion $\boldsymbol{\pi}$;

(4) Generate $\mathbf{x}_i$ according to the z_i-th Gaussian component.

The above process can be expressed as

$$\begin{aligned} \boldsymbol{\pi} &\sim \mathrm{GEM}(\alpha), && (1)\\ \boldsymbol{\theta}_k &\sim \mathrm{NIW}(\boldsymbol{\theta}_0), && (2)\\ z_i &\sim \mathrm{Multi}(\boldsymbol{\pi}), && (3)\\ \mathbf{x}_i &\sim \mathcal{N}(\boldsymbol{\theta}_{z_i}). && (4) \end{aligned}$$

Here GEM denotes the stick-breaking process, $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}(k = 1, 2, ..., \infty)$ is a set of parameters including the mean vector, $\boldsymbol{\mu}_k$, and covariance matrix, $\boldsymbol{\Sigma}_k$ of the k-th Gaussian component. $\boldsymbol{\theta}_0 = \{\mathbf{m}_0, \mathbf{S}_0, \kappa_0, \nu_0\}$ parameterizes the prior distribution in form of NIW where $\mathbf{m}_0$ is prior mean for $\boldsymbol{\mu}_k$, $\mathbf{S}_0$ is proportional to the prior mean for $\boldsymbol{\Sigma}_k$, κ_0 is the belief-strength in $\mathbf{m}_0$, and ν_0 is the belief-strength in $\mathbf{S}_0$.

2.2. Inference of DPGMM

Various algorithms [30–34] have been studied for inference of DPGMMs. Some of them [30, 31] are based on sampling using a Markov chain Monte Carlo (MCMC) scheme while others are based on variational inference [32–34]. In our work, we need an algorithm which explicitly represents the mixing weights $\boldsymbol{\pi}$ for the computation of GMM posteriorgrams, and can be highly parallelized so that the inference can be scalable to a huge amount of speech frames. Due to these requirements, we employ a parallelizable split-merge-based sampler [25]. It alternates between a restricted DPGMM Gibbs sampler and a set of split/merge moves to construct an exact MCMC sampling algorithm to conduct posterior sampling-based inference. The rest of this sub-section summarizes the procedures of the split-merge-based sampler.

(1) Restricted DPGMM Gibbs sampling. This part restricts z to be sampled only from the existing labels based on the fact that any realization of z belongs to a finite number of components. We denote the label assignment as $\mathcal{Z} = \{z_i\}_{i=1}^N$ where $z_i \in \{1, 2, ..., K\}(i = 1, 2, ..., N)$. Note that Dirichlet process (DP) has a property that the measure on any finite partitioning of the measurable space is distributed according to a Dirichlet distribution. As a result, the posterior sampler of $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_K, \pi'_{K+1})$ can be expressed as

$$(\pi_1, ..., \pi_K, \pi'_{K+1}) \sim \mathrm{Dir}(N_1, N_2, ..., N_K, \alpha), \quad (5)$$

where π'_{K+1} denotes the sum of all empty component weights and N_k is the number of observed data assigned with label k. α can be interpreted as the relative probability of assigning an observed data with a new component label. The sampling of $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ can be expressed as

$$\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \overset{\propto}{\sim} \mathrm{NIW}(\mathbf{m}_k, \mathbf{S}_k, \kappa_k, \nu_k), \forall k \in \{1, 2, ..., K\}, \quad (6)$$

where $a \overset{\propto}{\sim} b$ denotes sampling a from distribution proportional to b and the parameters of NIW are computed as follows:

$$\kappa_k = \kappa_0 + N_k, \nu_k = \nu_0 + N_k, \mathbf{m}_k = \frac{\kappa_0 \mathbf{m}_0 + N_k \bar{\mathbf{x}}_k}{\kappa_k},$$
$$\mathbf{S}_k = \mathbf{S}_0 + \sum_{\{i:z_i=k\}} \mathbf{x}_i \mathbf{x}_i^T + \kappa_0 \mathbf{m}_0 \mathbf{m}_0^T - \kappa_k \mathbf{m}_k \mathbf{m}_k^T,$$

where $\bar{\mathbf{x}}_k$ is the mean of $\{\mathbf{x}_i | z_i = k\}$. And z_i can be sampled as follows:

$$z_i \overset{\propto}{\sim} \sum_{k=1}^K \pi_k \mathcal{N}(\mathrm{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \mathbf{1}[z_i = k]. \quad (7)$$

where $\mathbf{1}[z_i = k]$ is a K-element vector whose z_i-th element equals 1 and others equal 0. Note that Eqs.(5)-(7) can be parallelized and compose a restricted DPGMM Gibbs sampler.

(2) Split/merge sampling. The previous part only samples labels from existing components so that it constructs a non-ergodic Markov Chain (MC) that one does not expect. Thus split/merge moves of the existing components emerge since it can form an exact ergodic MC. In the split/merge sampling procedure, there are two steps: (a) splitting each component into 2 sub-clusters to supply candidates for split moves; and (b) Metropolis-Hastings split/merge.

(2-a) Generating sub-clusters. Each component is split into 2 sub-clusters with mixing weights $\tilde{\boldsymbol{\pi}}_k = \{\tilde{\pi}_{k,l}, \tilde{\pi}_{k,r}\}$ and $\tilde{\boldsymbol{\theta}}_k = \{\tilde{\boldsymbol{\theta}}_{k,l}, \tilde{\boldsymbol{\theta}}_{k,r}\}$, and each observed data $\mathbf{x}_i$ is assigned with a sub-cluster label $\tilde{z}_i \in \{l, r\}$ indicating which sub-cluster it belongs to. The sampling is independent between different components and is parallelizable. To sample parameters of sub-clusters and sub-cluster label assignments, we use the following steps ($\forall k \in \{1, ..., K\}, \forall i \in \{1, ..., N\}, \forall s \in \{l, r\}$):

$$\tilde{\boldsymbol{\pi}}_k = (\tilde{\pi}_{k,l}, \tilde{\pi}_{k,r}) \sim \mathrm{Dir}(N_{k,l} + \alpha/2, N_{k,r} + \alpha/2), \quad (8)$$
$$\tilde{\boldsymbol{\theta}}_{k,s} \overset{\propto}{\sim} \mathcal{N}(\mathbf{x}_{k,s} | \tilde{\boldsymbol{\theta}}_{k,s}) \mathrm{NIW}(\tilde{\boldsymbol{\theta}}_{k,s} | \boldsymbol{\theta}_0), \quad (9)$$
$$\tilde{z}_i \overset{\propto}{\sim} \sum_{\{i:\tilde{z}_i=s\}} \tilde{\pi}_{z_i,s} \mathcal{N}(\mathbf{x}_i | \tilde{\boldsymbol{\theta}}_{z_i,s}). \quad (10)$$

where $N_{k,s}(s \in \{l, r\})$ is number of observed data assigned with sub-cluster label s in cluster z_i.

(2-b) Metropolis-Hastings split and merge. After the sub-cluster-related variables including $\tilde{\boldsymbol{v}}_k = \{\tilde{\boldsymbol{\pi}}_k, \tilde{\boldsymbol{\theta}}_k, \tilde{\mathcal{Z}}_k\}(\tilde{\mathcal{Z}}_k = \{\tilde{z}_i | z_i = k\}_{i=1}^N, k = 1, ..., K)$ are sampled according to Eqs.(8)-(10), we propose split or merge moves in a Metropolis-Hastings (MH) fashion. In the following description, the hat on the top of variables (e.g. $\pi, \boldsymbol{\theta}, \tilde{\boldsymbol{\pi}}, \tilde{\boldsymbol{\theta}}, \mathcal{Z}, \tilde{\mathcal{Z}}$) denotes the proposal for the variables. $Q \in \{Q_{split_c}, Q_{merge_{m,n}}\}$ denotes proposal move selected randomly from split move or merge move where Q_{split_c} denotes splitting component c into m and n, $Q_{merge_{m,n}}$ denotes merging components m and n into c. Conditioned on $Q = Q_{split_c}$, the proposed variables are sampled as follows:

$$(\hat{\mathcal{Z}}_m, \hat{\mathcal{Z}}_n) = \mathrm{split_c}(\mathcal{Z}, \tilde{\mathcal{Z}}), \quad (11)$$

$$(\hat{\pi}_m, \hat{\pi}_n) = \pi_c \boldsymbol{\pi}_{sub}, \boldsymbol{\pi}_{sub} = (\pi_m, \pi_n) \sim \mathrm{Dir}(\hat{N}_m, \hat{N}_n), \quad (12)$$

$$(\hat{\boldsymbol{\theta}}_m, \hat{\boldsymbol{\theta}}_n) \sim q(\hat{\boldsymbol{\theta}}_m, \hat{\boldsymbol{\theta}}_n | \mathcal{X}, \hat{\mathcal{Z}}, \hat{\tilde{\mathcal{Z}}}), \quad (13)$$

$$(\hat{\tilde{\boldsymbol{v}}}_m, \hat{\tilde{\boldsymbol{v}}}_n) \sim p(\hat{\tilde{\boldsymbol{v}}}_m, \hat{\tilde{\boldsymbol{v}}}_n | \mathcal{X}, \hat{\mathcal{Z}}), \quad (14)$$

and conditioned on $Q = Q_{merge_{m,n}}$, we propose samples as follows:

$$\hat{\mathcal{Z}}_c = \mathrm{merge_{m,n}}(\mathcal{Z}), \quad (15)$$

$$\hat{\pi}_c = \hat{\pi}_m + \hat{\pi}_n, \quad (16)$$

$$\hat{\boldsymbol{\theta}}_c \sim q(\hat{\boldsymbol{\theta}}_c | \mathcal{X}, \hat{\mathcal{Z}}, \hat{\tilde{\mathcal{Z}}}), \quad (17)$$

$$\hat{\tilde{\boldsymbol{v}}}_c \sim p(\hat{\tilde{\boldsymbol{v}}}_c | \mathcal{X}, \hat{\mathcal{Z}}). \quad (18)$$

In Eqs.(11)-(18), the function $\mathrm{split_c}(\cdot)$ splits the labels of component c according to the assignment of sub-clusters, $\mathrm{merge_{m,n}}(\cdot)$ merges labels of components m and n, $\hat{\mathcal{Z}}_k = \{z_i | z_i = k\}_{i=1}^N (k \in \{m, n, c\})$, and $\hat{N}_k (k \in \{m, n\})$ denotes the number of observed data labeled with k. Eqs.(13) and (17) proposing $\hat{\boldsymbol{\theta}}_k (k \in \{m, n, c\})$ are actually the same as Eqs.(6)-(7) and thus we simplify the distribution as q. To sample $\hat{\tilde{\boldsymbol{v}}}_k$ for a new proposed component from distribution $p(\cdot)$, we run a Gibbs sampler described in Eqs.(8)-(10). With the "Hastings ratio" H computed as suggested in [25], the proposed split/merge moves above are accepted with probability $\min\{1, H\}$ in an MH-MCMC framework. The Hastings ratio for merge moves above may decay sharply so that merge proposal is hardly accepted, thus a random merge sampler is employed to propose merge moves [25]. Note that, after the Hastings ratio determines the split/merge moves, sampling parameters of the new components can be parallelizable.

2.3. Generation of DPGMM Posteriorgrams

In our application, the observed data are speech frames $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$. DPGMM are inferred with K components together with their mixing weights, $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$, mean vectors, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ and covariance matrix, $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_k\}_{k=1}^K$. The posterior probability of the k-th component conditioned on i-th observed speech frame, $\mathbf{x}_i$, can be computed as follows:

$$p_{i,k} = p(c_k | \mathbf{x}_i) = \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \Sigma_j)}. \quad (19)$$

Then $P_i = (p_{i,1}, ..., p_{i,K})(i = 1, ..., N)$ forms a posteriorgram.

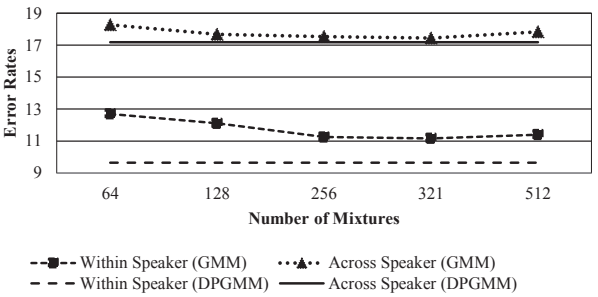

Figure 2: *Error rate (%) of ABX discrimination test on posteriorgrams of GMM with different numbers of components.*

3. Experiments

3.1. Corpus and Setup

To evaluate the effectiveness of our proposed features, experiments were conducted on the corpus provided by the zero resource speech challenge. This corpus consists of a 10 hour English dataset [35] and a 5 hour Xitsonga dataset [36]. Following the track 1 of the challenge, our evaluation metric is error rate in the ABX discriminability task [26, 27]. Supposing $\mathrm{S}(\mathbf{x})$ and $\mathrm{S}(\mathbf{y})$ are two sets of acoustic examples corresponding to category $\mathbf{x}$ and category $\mathbf{y}$, the correct rate (c) of ABX discrimination is calculated as follows:

$$c(\mathbf{x}, \mathbf{y}) = \frac{1}{m(m-1)n} \Sigma_{a \in \mathrm{S}(\mathbf{x})} \Sigma_{b \in \mathrm{S}(\mathbf{y})} \Sigma_{x \in \mathrm{S}(\mathbf{x}) \setminus \{a\}} \left(\delta_{\mathrm{d}(a,x) < \mathrm{d}(b,x)} + \frac{1}{2} \delta_{\mathrm{d}(a,x) = \mathrm{d}(b,x)}\right),$$

where m and n are the number of examples in $\mathrm{S}(\mathbf{x})$ and $\mathrm{S}(\mathbf{y})$, $\mathrm{d}(x, y)$ denotes the DTW divergence, δ is an indicator function. As suggested in [37], cosine distance was used for MFCC features, and KL-divergence was used for posteriorgrams to compute DTW divergences. The finally reported scores are the error rates of within-speaker and across-speaker ABX discrimination task where the correct rates are averaged over all found contexts for a given pair of central phonemes and then over all pairs of central phonemes [37].

We firstly removed the silence regions in the English utterances. Silence detection was not performed on the Xitsonga dataset because we observed that the Xitsonga dataset is relatively silence free. Then 39-dimensional MFCCs (13-dimensional MFCC+Δ+$\Delta\Delta$) were extracted with a 25ms analysis window and a 10ms window shift, and followed by mean and variance normalization (MVN) and vocal tract length normalization (VTLN). Stopped at the 1500-th iteration, two DPGMMs were trained with 8 cores of a workstation (Intel® Xeon®CPU W3520 @2.67GHz, 4GB memory). We consumed 9.33 hours and 6.52 hours for the English and Xitsonga datasets respectively. And then DPGMM-based posteriorgrams (DPGPG) are computed as the described in 2.3. The parameters including α and $\boldsymbol{\theta_0} = \{\mathbf{m}_0, \mathbf{S}_0, \kappa_0, \nu_0\}$ need manual setting in training. $\mathbf{m}_0$ and $\mathbf{S}_0$ are set as the global mean and covariance of the post-processed MFCCs, respectively. ν_0 is set to 41 as the common choice suggested in [38]. We conducted an initial study of the influences of α and κ_0 illustrated at Figure 3. Unless stated otherwise, α and κ_0 are set to 1.

We also compared our proposed features with MFCC and posteriorgrams derived from language-mismatched supervised phoneme recognizers. Posteriorgrams derived from language-mismatched phoneme recognizers have widely been used in a typical zero-resource task, such as query-by-example

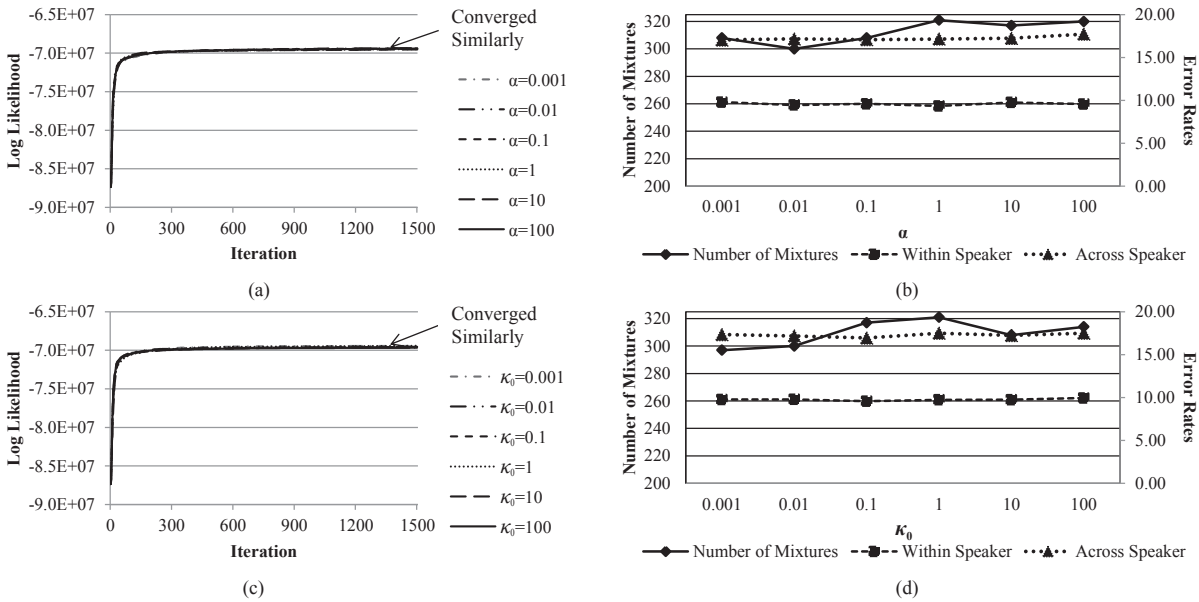

Figure 3: *Study of performances over α and κ_0 on Xitsonga dataset. (a)-(b): Performance over α; (c)-(d): Performance over κ_0.*

spoken term detection. In our experiment, phoneme state-posteriorgrams from BUT phoneme recognizer [39] for Czech (CZPG), Hungarian (HUPG) and Russian (RUPG) were evaluated together with a fusion version of them (FUPG).

3.2. Results and Discussion

Table 1: *Error rates (%) of ABX discrimination test.*

Feature	English			Xitsonga		
	NDim	Within (%)	Across (%)	NDim	Within (%)	Across (%)
MFCC	39	17.2	26.8	39	19.6	30.8
DPGPG	385	10.8	16.3	321	9.6	17.2
CZPG	138	11.4	17.2	138	11.8	16.8
HUPG	186	11.1	16.5	186	11.7	17.2
RUPG	159	11.7	17.3	159	11.3	15.6
FUPG	483	10.4	15.8	483	12.0	15.5
DPGPG+ FUPG	868	**9.7**	**14.9**	804	**9.5**	**15.0**

Tabel 1 summarizes the evaluation results in both within-speaker and across-speaker tests together with the dimension (NDim) of acoustic features. Our results of MFCC slightly differ from the baseline in [37] probably due to that our MFCC is 39-dimensional with Δ and $\Delta\Delta$. As illustrated in Table 1, DPGMMs inferred 385 and 321 components on the English and Xitsonga dataset respectively. More components being inferred in the English dataset is possibly because of more diversified speech characteristics due to less strict recording condition. Compared with each kind of phoneme state posteriorgrams (CZPG, HUPG and RUPG), DPGPG obtained the lowest within-speaker error rate on both datasets. In terms of the across-speaker error rate, DPGPG performed comparably to each kind of phoneme state posteriorgrams, and DPGPG even outperformed each of them on the English dataset. DPGPG also showed comparable performance to FUPG. We believe that different kinds of posteriorgrams characterize the test utterances in different aspects and carry complementary information, and the lowest error rates are obtained in the two datasets through the fusion of features (DPGPG+FUPG).

We trained a set of parametric GMMs with different number of components on the Xitsonga dataset using the voicebox toolkit [40]. The error rates of the corresponding GMM-based posteriorgrams together with that of DPGMM posteriorgrams are plotted in Figure 2. We observed that DPGMM can learn a proper number of components.

Figure 3 shows the effect of different values of α and κ_0. Figure 3.(a) and (c) plot the log-likelihood curves against the number of iterations when different values of α and κ_0 are used. We observed that convergence of DPGMM is stable against different hyper-parameter values. Figure 3.(b) and (d) show that the number of inferred mixture components varies in a small range between 300 and 321, and between 295 and 321 when altering the values of α and κ_0 respectively. The posteriorgrams extracted on different settings of hyper-parameters give similar results of within-speaker and across-speaker error rates. Illustrations above indicate that we can get stable DPGMM posteriorgrams with little worry about the influence of hyper-parameters.

4. Conclusions

We adopt a parallel sampler for the inference of DPGMMs. The restricted Gibbs sampler with proposed split/merge moves facilitates parallelization, which leads to an efficient inference algorithm which is scalable to a large amount of speech frames. Our experiments show that DPGMM can determine the best number of Gaussian components itself without parameter tuning in a development set, and the inference of DPGMM is insensitive to the choice of the hyper-parameters. This is particularly suitable for low-resource scenarios. Moreover, the DPGMM posteriorgrams can perform comparably to the phoneme state posteriorgrams derived from language-mismatched phoneme recognizers.

5. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61175018) and COLIPS.

6. References

[1] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 16, no. 1, pp. 186–197, 2008.

[2] Y. Zhang and J. R. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *Proc. of ICASSP*. IEEE, 2010, pp. 4366–4369.

[3] A. Muscariello, G. Gravier, and F. Bimbot, "Unsupervised motif acquisition in speech via seeded discovery and template matching combination," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 20, no. 7, pp. 2031–2044, 2012.

[4] M. Huijbregts, M. McLaren, and D. Van Leeuwen, "Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection," in *Proc. of ICASSP*. IEEE, 2011, pp. 4436–4439.

[5] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 15, no. 1, pp. 271–284, 2007.

[6] B. Varadarajan, S. Khudanpur, and E. Dupoux, "Unsupervised learning of acoustic sub-word units," in *Proc. ACL-HLT*. ACL, 2008, pp. 165–168.

[7] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. of ASRU*. IEEE, 2009, pp. 421–426.

[8] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. of ICASSP*. IEEE, 2012, pp. 5157–5160.

[9] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training." in *Proc. of ICASSP*. IEEE, 2013, pp. 8091–8095.

[10] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. of ICASSP*. IEEE, 2014, pp. 7634–7638.

[11] G. S. T. Schatz and E. Dupoux, "Phonetics embedding learning with side information," in *Proc. of SLT*. IEEE, 2014.

[12] M.-H. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, "Unsupervised training of an HMM-based self-organizing unit recognizer with applications to topic classification and keyword discovery," *Computer Speech & Language*, vol. 28, no. 1, pp. 210–223, 2014.

[13] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *IEEE/ACM Trans. Audio, Speech, and Language Process.*, vol. 23, no. 2, pp. 264–277, Feb 2015.

[14] I. Malioutov, A. Park, R. Barzilay, and J. Glass, "Making sense of sound: Unsupervised topic segmentation over acoustic input," *ACL 2007*, pp. 504–511, 2007.

[15] L. Zheng, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Acoustic texttiling for story segmentation of spoken documents," in *Proc. of ICASSP*. IEEE, 2012, pp. 5121–5124.

[16] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. of ASRU*. IEEE, 2009, pp. 398–403.

[17] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, "NLP on spoken documents without asr," in *Proc. of EMNLP*. ACL, 2010, pp. 460–470.

[18] X. Zhu, G. Penn, and F. Rudzicz, "Summarizing multiple spoken documents: finding evidence from untranscribed audio," in *Proc. of ACL-IJCNLP*. ACL, 2009, pp. 549–557.

[19] C.-Y. Lee and J. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proc. of ACL*. ACL, 2012, pp. 40–49.

[20] A. Jansen, S. Thomas, and H. Hermansky, "Intrinsic spectral analysis for zero and high resource speech recognition." in *Proc. of INTERSPEECH*, 2012, pp. 879–882.

[21] A. Jansen and P. Niyogi, "Intrinsic spectral analysis," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1698–1710, 2013.

[22] P. Yang, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Intrinsic spectral analysis based on temporal context features for query-by-example spoken term detection," in *Proc. of INTERSPEECH*, 2014, pp. 1722–1726.

[23] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. of ICASSP*. IEEE, 2015.

[24] H. Kamper, A. Jansen, S. King, and S. Goldwater, "Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings," in *Proc. of SLT*. IEEE, 2014.

[25] J. Chang and J. W. Fisher III, "Parallel sampling of DP mixture models using sub-clusters splits," in *Proc. of NIPS*, 2013, pp. 620–628.

[26] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux, "Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline," in *Proc. of INTERSPEECH*, 2013, pp. 1–5.

[27] T. Schatz, V. Peddinti, X.-N. Cao, F. Bach, H. Hermansky, and E. Dupoux, "Evaluating speech features with the Minimal-Pair ABX task (ii): Resistance to noise," in *Proc. of INTERSPEECH*, 2014, pp. 915–919.

[28] J. Sethuraman, "A constructive definition of Dirichlet priors," DTIC Document, Tech. Rep., 1991.

[29] M. West and J. Harrison, "Multivariate modelling and forecasting," in *Bayesian Forecasting and Dynamic Models*, ser. Springer Series in Statistics. Springer New York, 1997, pp. 581–630.

[30] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *J. computational and graphical statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[31] S. Jain and R. M. Neal, "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model," *J. Computational and Graphical Statistics*, vol. 13, no. 1, 2004.

[32] D. M. Blei, M. I. Jordan *et al.*, "Variational inference for Dirichlet process mixtures," *Bayesian analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[33] K. Kurihara, M. Welling, and N. A. Vlassis, "Accelerated variational Dirichlet process mixtures," in *proc. of NIPS*, 2006, pp. 761–768.

[34] K. Kurihara, M. Welling, and Y. W. Teh, "Collapsed variational Dirichlet process mixture models." in *Proc. of IJCAI*, vol. 7, 2007, pp. 2796–2801.

[35] M. A. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, "Buckeye corpus of conversational speech (2nd release)[www. buckeyecorpus. osu. edu] Columbus, OH: Department of Psychology," *Ohio State University (Distributor)*, 2007.

[36] N. J. De Vries, M. H. Davel, J. Badenhorst, W. D. Basson, F. De Wet, E. Barnard, and A. De Waal, "A smartphone-based asr data collection tool for under-resourced languages," *Speech Commun.*, vol. 56, pp. 119–131, 2014.

[37] http://www.lscp.net/persons/dupoux/bootphon/zerospeech2014/website/.

[38] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[39] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme," in *Proc. of ICASSP*. IEEE, 2006, pp. 325–328.

[40] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.