8 Markov chain Monte Carlo

For most probabilistic models of practical interest, exact inference is intractable, and so we have to resort to some form of approximation. Markov chain Monte Carlo (MCMC) is another realization of full Bayesian treatment of practical Bayesian solutions (Neal 1993, Gilks, Richardson & Spiegelhalter 1996, Bishop 2006). MCMC is known as a *stochastic* approximation which acts differently from the *deterministic* approximation based on variational Bayes (VB) as addressed in Chapter 7. Variational inference using VB approximates the posterior distribution through factorization of the distribution over multiple latent variables and scales well to large applications. MCMC uses the numerical sampling computation rather than solving integrals and expectation analytically. Since MCMC can use any distributions in principle, it is capable of wide applications, and can be used for Bayesian nonparametric (BNP) learning, which provides highly flexible models whose complexity grows appropriately with the amount of data. Although, due to the computational cost, the application of MCMC to speech and language processing is limited to small-scale problems currently, this chapter describes promising new directions of Bayesian nonparametrics for speech and language processing by automatically growing models to deal with speaker diarization, acoustic modeling, language acquisition, and hierarchical language modeling. The strengths and weaknesses using VB and MCMC are complementary. In what follows, we first introduce the general background of sampling methods including MCMC and Gibbs sampling algorithms. Next, the Bayesian nonparametrics are calculated to build a flexible topic model based on the hierarchical Dirichlet process (HDP) (Teh *et al*. 2006). Several applications in speech and language processing areas are surveyed. GMM-based speaker clustering, CDHMM-based acoustic unit discovery by using MCMC, and the language model based on the hierarchical Pitman–Yor process (Teh *et al*. 2006) are described.

The fundamental problem in MCMC involves finding the expectation of some function $f(\boldsymbol{\theta})$ with respect to a probability distribution $p(\boldsymbol{\theta})$ where the components of $\boldsymbol{\theta}$ might comprise discrete or continuous variables, which are some factors or parameters to be inferred under a probabilistic model. In the case of continuous variables, we would like to evaluate the expectation

$$\mathbb{E}_{(\boldsymbol{\theta})}[f(\boldsymbol{\theta})] = \int f(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{8.1}$$

where the integral is replaced by summation in the case of discrete variables. We assume that such expectations are too complicated to be evaluated analytically. The general

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

idea behind sampling methods is to obtain a set of samples $\{\boldsymbol{\theta}^{(l)}, l = 1, \cdots, L\}$ drawn independently from the distribution $p(\boldsymbol{\theta})$. We may approximate the integral by a sample mean of function $f(\boldsymbol{\theta})$ over these samples $\boldsymbol{\theta}^{(l)}$:

$$\widehat{f} = \frac{1}{L} \sum_{l=1}^{L} f(\boldsymbol{\theta}^{(l)}). \tag{8.2}$$

Since the samples $\boldsymbol{\theta}^{(l)}$ are drawn from the distribution $p(\boldsymbol{\theta})$, the estimator $\widehat{f}$ has the correct mean, i.e., $\widehat{f} = \mathbb{E}_{(\boldsymbol{\theta})}[f(\boldsymbol{\theta})]$. In general, ten to twenty independent samples may suffice to estimate an expectation. However, the samples $\{\boldsymbol{\theta}^{(l)}\}$ may not be drawn independently. The effective sample size might be much smaller than the apparent sample size L. This implies that a relatively large sample size is required to achieve sufficient accuracy.

8.1 Sampling methods

A simple strategy can be designed to generate random samples from a given distribution. We first consider how to generate random numbers from non-uniform distributions, assuming that we already have a source of uniformly distributed random numbers. Let θ be uniformly distributed by $p(\theta) = 1$ over the interval $(0, 1)$. We transform the values of θ using some function $f(\cdot)$ by $y = f(\theta)$. The distributions of variables θ and y are related by

$$p(y) = p(\theta) \left| \frac{d\theta}{dy} \right|. \tag{8.3}$$

Taking integrals for both sides of Eq. (8.3), we have

$$\theta = h(y) = \int_{-\infty}^{y} p(\widetilde{y}) d\widetilde{y}, \tag{8.4}$$

which is the indefinite integral of $p(y)$. Thus, $y = h^{-1}(\theta)$, meaning that we have to transform the uniformly distributed random numbers θ using a function $h^{-1}(\cdot)$, which is the inverse of the indefinite integral of the desired distribution of y. Figure 8.1 depicts the geometrical interpretation of the transformation method for generating non-uniformly distributed random numbers. In addition, the generalization to multiple variables is straightforward and involves the Jacobian of the transform of variables:

$$p(y_1, \cdots, y_M) = p(\theta_1, \cdots, \theta_M) \left| \frac{\partial(\theta_1, \cdots, \theta_M)}{\partial(y_1, \cdots, y_M)} \right|. \tag{8.5}$$

A similar scheme can be applied to draw a multivariate distribution with M variables.

8.1.1 Importance sampling

The technique of *importance sampling* provides a framework for approximating the expectation in Eq. (8.1) directly but does not provide the mechanism for drawing samples from distribution $p(\boldsymbol{\theta})$. The finite sum approximation to expectation in Eq. (8.2)

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

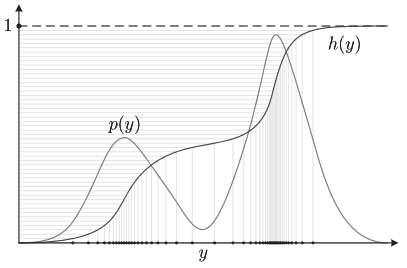


Figure 8.1 Transformation method for generating non-uniformly distributed random numbers from probability distribution $p(y)$. Function $h(y)$ represents the indefinite integral of $p(y)$. Adapted from Bishop (2006).

depends on being able to draw samples from the distribution $p(\boldsymbol{\theta})$. One simple strategy for evaluating expectation function would be to discretize $\boldsymbol{\theta}$-space into a uniform grid, and to evaluate the integrand as a sum of the form

$$\mathbb{E}_{(\boldsymbol{\theta})}[f(\boldsymbol{\theta})] \simeq \sum_{l=1}^{L} p(\boldsymbol{\theta}^{(l)}) f(\boldsymbol{\theta}^{(l)}). \tag{8.6}$$

However, the problem with this approach is that the number of terms in the summation grows exponentially with the dimensionality of $\boldsymbol{\theta}$. In many cases, the probability distributions of interest often have much of their mass confined to relatively small regions of $\boldsymbol{\theta}$ space. Accordingly, uniform sampling is very inefficient because in high-dimensional space, only a very small proportion of the samples make a significant contribution to the sum. We would like to sample the points falling in regions where $p(\boldsymbol{\theta})$ is large, or where the product $p(\boldsymbol{\theta})f(\boldsymbol{\theta})$ is large.

Suppose we wish to sample from a distribution $p(\boldsymbol{\theta})$ that is not simple or a standard distribution. Sampling directly from $p(\boldsymbol{\theta})$ is difficult. Importance sampling is based on the use of a *proposal distribution* $q(\boldsymbol{\theta})$ from which it is easy to draw samples. Figure 8.2 illustrates the proposal distribution for importance sampling. The expectation in Eq. (8.1) is expressed in the form of a finite sum over samples $\{\boldsymbol{\theta}^{(l)}\}$ drawn from $q(\boldsymbol{\theta})$:

$$\begin{aligned}\mathbb{E}_{(\boldsymbol{\theta})}[f(\boldsymbol{\theta})] &= \int f(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \int f(\boldsymbol{\theta}) \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &\simeq \frac{1}{L} \sum_{l=1}^{L} \frac{p(\boldsymbol{\theta}^{(l)})}{q(\boldsymbol{\theta}^{(l)})} f(\boldsymbol{\theta}^{(l)}).\end{aligned} \tag{8.7}$$

The quantities $r_l = p(\boldsymbol{\theta}^{(l)})/q(\boldsymbol{\theta}^{(l)})$ are known as the *importance weights*, and they correct the bias introduced by sampling from the wrong distribution.

Usually, the distribution $p(\boldsymbol{\theta})$ can only be evaluated up to a normalization constant, so that $p(\boldsymbol{\theta}) = \widetilde{p}(\boldsymbol{\theta})/Z_p$, where $\widetilde{p}(\boldsymbol{\theta})$ can be evaluated easily and Z_p is unknown. We may use the importance sampling distribution $q(\boldsymbol{\theta}) = \widetilde{q}(\boldsymbol{\theta})/Z_q$ to determine the expectation function:

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

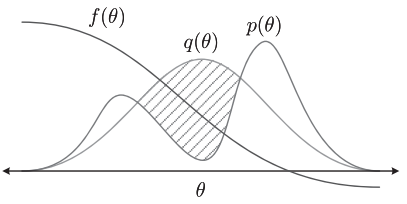


Figure 8.2 Proposal distribution $q(\theta)$ for importance sampling in estimation of expectation of function $f(\theta)$ with probability distribution $p(\theta)$. Adapted from Bishop (2006).

$$\begin{aligned}\mathbb{E}_{(\boldsymbol{\theta})}[f(\boldsymbol{\theta})] &= \frac{Z_q}{Z_p}\int f(\boldsymbol{\theta})\frac{\widetilde{p}(\boldsymbol{\theta})}{\widetilde{q}(\boldsymbol{\theta})}q(\boldsymbol{\theta})d\boldsymbol{\theta} \\ &\simeq \frac{Z_q}{Z_p}\frac{1}{L}\sum_{l=1}^{L}\widetilde{r}_l f(\boldsymbol{\theta}^{(l)}),\end{aligned} \tag{8.8}$$

where $\widetilde{r}_l = \widetilde{p}(\boldsymbol{\theta}^{(l)})/\widetilde{q}(\boldsymbol{\theta}^{(l)})$. We have

$$\begin{aligned}\frac{Z_p}{Z_q} &= \frac{1}{Z_q}\int \widetilde{p}(\boldsymbol{\theta})d\boldsymbol{\theta} = \int \frac{\widetilde{p}(\boldsymbol{\theta})}{\widetilde{q}(\boldsymbol{\theta})}q(\boldsymbol{\theta})d\boldsymbol{\theta} \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\widetilde{r}_l.\end{aligned} \tag{8.9}$$

We then obtain

$$\mathbb{E}_{(\boldsymbol{\theta})}[f(\boldsymbol{\theta})] \simeq \sum_{l=1}^{L} w_l f(\boldsymbol{\theta}^{(l)}), \tag{8.10}$$

where we define

$$w_l = \frac{\widetilde{r}_l}{\sum_m \widetilde{r}_m} = \frac{\widetilde{p}(\boldsymbol{\theta}^{(l)})/q(\boldsymbol{\theta}^{(l)})}{\sum_m \widetilde{p}(\boldsymbol{\theta}^{(m)})/q(\boldsymbol{\theta}^{(m)})}. \tag{8.11}$$

Clearly, the performance of the importance sampling method highly depends on how well the sampling distribution $q(\boldsymbol{\theta})$ matches the desired distribution $p(\boldsymbol{\theta})$.

8.1.2 Markov chain

One major weakness in evaluation of expectation function based on the importance sampling strategy is the severe limitation in spaces of high dimensionality. We accordingly turn to a very general and powerful framework called Markov chain Monte Carlo (MCMC), which allows sampling from a large class of distributions and which scales well with the dimensionality of the sample space. Before discussing MCMC methods in more detail, it is useful to study some general properties of Markov chains and investigate under what conditions a Markov chain can converge to the desired distribution. A

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

first-order Markov chain is defined for a series of variables $\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(M)}$ such that the following conditional independence holds for $m \in \{1, \cdots, M-1\}$:

$$p(\boldsymbol{\theta}^{(m+1)}|\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(m)}) = p(\boldsymbol{\theta}^{(m+1)}|\boldsymbol{\theta}^{(m)}). \tag{8.12}$$

This Markov chain starts from the probability distribution for initial variable $p(\boldsymbol{\theta}^{(0)})$ and operates with the transition probability $p(\boldsymbol{\theta}^{(m+1)}|\boldsymbol{\theta}^{(m)})$. A Markov chain is called homogeneous if the transition probabilities are unchanged for all m. The marginal probability for a variable $\boldsymbol{\theta}^{(m+1)}$ is expressed in terms of the marginal probabilities over the previous variable $\{\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(m)}\}$ in the chain,

$$p(\boldsymbol{\theta}^{(m+1)}) = \sum_{\boldsymbol{\theta}^{(m)}} p(\boldsymbol{\theta}^{(m+1)}|\boldsymbol{\theta}^{(m)}) p(\boldsymbol{\theta}^{(m)}). \tag{8.13}$$

A distribution is said to be *invariant* or stationary with respect to a Markov chain if each step in the chain keeps the distribution invariant. For a homogeneous Markov chain with transition probability $T(\boldsymbol{\theta}', \boldsymbol{\theta})$, the distribution $p^\star(\boldsymbol{\theta})$ is invariant if it has the following property:

$$p^\star(\boldsymbol{\theta}) = \sum_{\boldsymbol{\theta}'} T(\boldsymbol{\theta}', \boldsymbol{\theta}) p^\star(\boldsymbol{\theta}'). \tag{8.14}$$

Our goal is to use Markov chains to sample from a given distribution. We can achieve this goal if we set up a Markov chain such that the desired distribution is invariant. It is required that for $m \to \infty$, the distribution $p(\boldsymbol{\theta}^{(m)})$ converges to the required invariant distribution $p^\star(\boldsymbol{\theta})$, which is obtained irrespective of the choice of initial distribution $p(\boldsymbol{\theta}^{(0)})$. This invariant distribution is also called the *equilibrium* distribution. A sufficient condition for an invariant distribution $p(\boldsymbol{\theta})$ is to choose the transition probabilities to satisfy the *detailed balance*, i.e.

$$p^\star(\boldsymbol{\theta}) T(\boldsymbol{\theta}, \boldsymbol{\theta}') = p^\star(\boldsymbol{\theta}') T(\boldsymbol{\theta}', \boldsymbol{\theta}), \tag{8.15}$$

for a particular distribution $p^\star(\boldsymbol{\theta})$.

8.1.3 The Metropolis–Hastings algorithm

As discussed in importance sampling, we keep sampling from a proposal distribution and maintain a record of the current state $\boldsymbol{\theta}^{(\tau)}$. The proposal distribution $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(\tau)})$ depends on this current state. The sequence of samples $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \cdots$ forms a Markov chain. The proposal distribution is chosen to be sufficiently simple to draw samples directly. At each sampling cycle, we generate a candidate sample $\boldsymbol{\theta}^\star$ from the proposal distribution and then accept the sample according to an appropriate criterion. In a basic Metropolis algorithm (Metropolis, Rosenbluth, Rosenbluth *et al.* 1953), the proposal distribution is assumed to be symmetric, namely $q(\boldsymbol{\theta}_a|\boldsymbol{\theta}_b) = q(\boldsymbol{\theta}_b|\boldsymbol{\theta}_a)$ for all values of $\boldsymbol{\theta}_a$ and $\boldsymbol{\theta}_b$. The candidate sample is accepted with the probability

$$A(\boldsymbol{\theta}^\star, \boldsymbol{\theta}^{(\tau)}) = \min\left(1, \frac{\widetilde{p}(\boldsymbol{\theta}^\star)}{\widetilde{p}(\boldsymbol{\theta}^{(\tau)})}\right), \tag{8.16}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

where $p(\boldsymbol{\theta}) = \widetilde{p}(\boldsymbol{\theta})/Z_p$ with a readily evaluated distribution $\widetilde{p}(\boldsymbol{\theta})$ and an unknown normalization value Z_p. To fulfil this algorithm, we can choose a random number u with uniform distribution over the unit interval $(0, 1)$ and accept the sample if $A(\boldsymbol{\theta}^\star, \boldsymbol{\theta}^{(\tau)}) > u$. Definitely, if the step from $\boldsymbol{\theta}^{(\tau)}$ to $\boldsymbol{\theta}^\star$ causes an increase in the value of $p(\boldsymbol{\theta})$, then the candidate point is accepted. Once the candidate sample is accepted, then $\boldsymbol{\theta}^{(\tau+1)} = \boldsymbol{\theta}^\star$, otherwise the candidate sample $\boldsymbol{\theta}^\star$ is discarded, $\boldsymbol{\theta}^{(\tau+1)}$ is set to $\boldsymbol{\theta}^{(\tau)}$. The next candidate sample is drawn from the distribution $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(\tau+1)})$. This leads to multiple copies of samples in the final list of samples. As long as $q(\boldsymbol{\theta}_a|\boldsymbol{\theta}_b)$ is positive for any values of $\boldsymbol{\theta}_a$ and $\boldsymbol{\theta}_b$, the distribution of $\boldsymbol{\theta}^{(\tau)}$ approaches $p(\boldsymbol{\theta})$ as $\tau \to \infty$.

The basic Metropolis algorithm is further generalized to the Metropolis–Hastings algorithm (Hastings 1970) which is widely adopted in MCMC inference. This generalization is developed by relaxing the assumption in the Metropolis algorithm that the proposal distribution is no longer a symmetric function of its arguments. Using this algorithm, at step τ with current state $\boldsymbol{\theta}^{(\tau)}$, we draw a sample $\boldsymbol{\theta}^\star$ from the proposal distribution $q_k(\boldsymbol{\theta}|\boldsymbol{\theta}^{(\tau)})$ and then accept it with the probability

$$A_k(\boldsymbol{\theta}^\star, \boldsymbol{\theta}^{(\tau)}) = \min\left(1, \frac{\widetilde{p}(\boldsymbol{\theta}^\star) q_k(\boldsymbol{\theta}^{(\tau)}|\boldsymbol{\theta}^\star)}{\widetilde{p}(\boldsymbol{\theta}^{(\tau)}) q_k(\boldsymbol{\theta}^\star|\boldsymbol{\theta}^{(\tau)})}\right), \tag{8.17}$$

where k denotes the members of the set of possible transitions. For the case of a symmetric proposal distribution, the Metropolis–Hastings criterion in Eq. (8.17) is reduced to the Metropolis criterion in Eq. (8.16). We can show that $p(\boldsymbol{\theta})$ is an invariant distribution of the Markov chain generated by the Metropolis–Hastings algorithm by investigating the property of the detailed balance in Eq. (8.15). We find that

$$\begin{aligned} p(\boldsymbol{\theta}) q_k(\boldsymbol{\theta}|\boldsymbol{\theta}') A_k(\boldsymbol{\theta}', \boldsymbol{\theta}) &= \min(p(\boldsymbol{\theta}) q_k(\boldsymbol{\theta}|\boldsymbol{\theta}'), p(\boldsymbol{\theta}') q_k(\boldsymbol{\theta}'|\boldsymbol{\theta})) \\ &= \min(p(\boldsymbol{\theta}') q_k(\boldsymbol{\theta}'|\boldsymbol{\theta}), p(\boldsymbol{\theta}) q_k(\boldsymbol{\theta}|\boldsymbol{\theta}')) \\ &= p(\boldsymbol{\theta}') q_k(\boldsymbol{\theta}'|\boldsymbol{\theta}) A_k(\boldsymbol{\theta}, \boldsymbol{\theta}'). \end{aligned} \tag{8.18}$$

The choice of proposal distribution is important in an MCMC algorithm. For continuous state spaces, a common choice is a Gaussian centered on the current state, leading to an important trade-off in determining the variance parameter of this distribution. If the variance is small, the proportion of accepted transitions is high, but a slow random walk is taken through the state space. On the other hand, if the variance parameter is large, the rejection rate is high because the updated state has low probability $p(\boldsymbol{\theta})$. Figure 8.3 shows a schematic for selecting an isotropic Gaussian proposal distribution to sample random numbers from a correlated multivariate Gaussian distribution. In order to keep the rejection rate low, the scale ρ of the proposal distribution $q_k(\boldsymbol{\theta}|\boldsymbol{\theta}^{(\tau)})$ should be comparable to the smallest standard deviation $\sigma_{\min}$, which leads to a random walk so that the number of steps for separating states is of order $(\sigma_{\max}/\sigma_{\min})^2$ where $\sigma_{\max}$ is the largest standard deviation.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

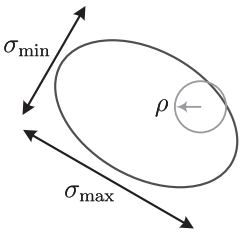


Figure 8.3 Using an isotropic Gaussian proposal distribution (circle) to sample random numbers for a correlated bivariate Gaussian distribution (ellipse). Adapted from Bishop (2006).

8.1.4 Gibbs sampling

Gibbs sampling (Geman & Geman 1984, Liu 2008) is a simple and widely applicable realization of an MCMC algorithm which is a special case of the Metropolis–Hastings algorithm. Consider the distribution of M random variables $p(\boldsymbol{\theta}) = p(\theta_1, \cdots, \theta_M)$ and suppose that we have some initial state for the Markov chain. Each step of the Gibbs sampling procedure replaces the value of one of the variables by a value drawn from the distribution of that variable conditioned on the values of the remaining states. That is to say, we replace the ith component θ_i by a value drawn from the distribution $p(\theta_i|\boldsymbol{\theta}^{-i})$, where $\boldsymbol{\theta}^{-i}$ denotes $\theta_1, \cdots, \theta_M$ but with θ_i omitted. The sampling procedure is repeated by cycling through the variables in a particular order or in a random order from some distribution. This procedure samples the required distribution $p(\boldsymbol{\theta})$, which should be invariant at each of the Gibbs sampling steps or in the whole Markov chain. This is because the marginal distribution $p(\boldsymbol{\theta}^{-i})$ is invariant and the conditional distribution $p(\theta_i|\boldsymbol{\theta}^{-i})$ is correct at each sampling step. Gibbs sampling of M variables for T steps follows this procedure:

- Initialize $\{\theta_i : i = 1, \cdots, M\}$.
- For $\tau = 1, \cdots, T$:
  - Sample $\theta_1^{(\tau+1)} \sim p(\theta_1|\theta_2^{(\tau)}, \theta_3^{(\tau)}, \cdots, \theta_M^{(\tau)})$.
  - Sample $\theta_2^{(\tau+1)} \sim p(\theta_2|\theta_1^{(\tau+1)}, \theta_3^{(\tau)}, \cdots, \theta_M^{(\tau)})$.
  - $\vdots$
  - Sample $\theta_j^{(\tau+1)} \sim p(\theta_j|\theta_1^{(\tau+1)}, \cdots, \theta_{j-1}^{(\tau+1)}, \theta_{j+1}^{(\tau)}, \cdots, \theta_M^{(\tau)})$.
  - $\vdots$
  - Sample $\theta_M^{(\tau+1)} \sim p(\theta_M|\theta_1^{(\tau+1)}, \theta_2^{(\tau+1)}, \cdots, \theta_{M-1}^{(\tau+1)})$.

Gibbs sampling can be shown to be a special case of the Metropolis–Hastings algorithm. Consider the Metropolis–Hastings sampling step involving the variable θ_k in which the remaining variables $\boldsymbol{\theta}^{-k}$ are fixed. The transition probability from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}^\star$ is then given by $q_k(\boldsymbol{\theta}^\star|\boldsymbol{\theta}) = p(\theta_k^\star|\boldsymbol{\theta}^{-k})$, because the remaining variables are unchanged by the sampling step, and $(\boldsymbol{\theta}^\star)^{-k} = \boldsymbol{\theta}^{-k}$. By using $p(\boldsymbol{\theta}) = p(\theta_k|\boldsymbol{\theta}^{-k})p(\boldsymbol{\theta}^{-k})$, the acceptance probability in the Metropolis–Hastings algorithm is obtained by

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

$$A(\boldsymbol{\theta}^\star, \boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}^\star) q_k(\boldsymbol{\theta}|\boldsymbol{\theta}^\star)}{p(\boldsymbol{\theta}) q_k(\boldsymbol{\theta}^\star|\boldsymbol{\theta})}$$
$$= \frac{p(\theta_k^\star|(\boldsymbol{\theta}^\star)^{-k}) p((\boldsymbol{\theta}^\star)^{-k}) p(\theta_k|(\boldsymbol{\theta}^\star)^{-k})}{p(\theta_k|\boldsymbol{\theta}^{-k}) p(\boldsymbol{\theta}^{-k}) p(\theta_k^\star|\boldsymbol{\theta}^{-k})} = 1, \tag{8.19}$$

where $(\boldsymbol{\theta}^\star)^{-k} = \boldsymbol{\theta}^{-k}$ is applied. This result indicates that the sampling steps in the Metropolis–Hastings algorithm are always accepted.

Collapsed Gibbs sampling

Collapsed Gibbs sampling (Liu 1994, Griffiths & Steyvers 2004) is a method for using a marginal conditional distribution where some of variables are integrated out, instead of sampling. For example, suppose we have a set of variables Λ, the proposal distribution in the Gibbs sampling is represented as follows:

$$p(\theta_i|\boldsymbol{\theta}^{-i}) = \int p(\theta_i|\boldsymbol{\theta}^{-i}, \Lambda) p(\Lambda) d\Lambda. \tag{8.20}$$

$p(\Lambda)$ is a prior distribution. This integral can be analytically solved when we use a conjugate prior, and the following sections sometimes use a marginal conditional distribution.

8.1.5 Slice sampling

One weakness in the Metropolis–Hastings algorithm is the sensitivity to step size. If this is too small, the result has slow decorrelation due to random walk behavior, while if it is too large, the sampling procedure is not efficient due to a high rejection rate. The *slice sampling* approach (Neal 2003) provides an adaptive step size which is automatically adjusted to fit the characteristics of the distribution. Again, it is required that the unnormalized distribution $\widetilde{p}(\boldsymbol{\theta})$ is available to be evaluated. Consider the univariate case. Slice sampling is performed by augmenting θ with an additional variable u and drawing samples from the joint space of (θ, u). The goal is to sample uniformly from the area under the distribution given by

$$\widetilde{p}(\theta, u) = \begin{cases} 1/Z_p & \text{if } 0 \leqslant u \leqslant \widetilde{p}(\theta), \\ 0 & \text{otherwise,} \end{cases} \tag{8.21}$$

where $Z_p = \int \widetilde{p}(\theta) d\theta$. The marginal distribution of θ is given by

$$\int \widetilde{p}(\theta, u) du = \int_0^{\widetilde{p}(\theta)} \frac{1}{Z_p} du = \frac{\widetilde{p}(\theta)}{Z_p} = p(\theta). \tag{8.22}$$

To carry out this scheme, we first sample from $p(\theta)$ by sampling from $\widetilde{p}(\theta, u)$ and then neglecting the u values. Alternatively sampling θ and u can be achieved. Given the value of θ, we evaluate $\widetilde{p}(\theta)$ and then sample u uniformly in the range $0 \leqslant u \leqslant \widetilde{p}(\theta)$. We then fix u and sample θ uniformly from the "slice" through the distribution defined by $\{\theta : \widetilde{p}(\theta) > u\}$. As illustrated in Figure 8.4(a), for a given value $\theta^{(\tau)}$, a value of u is chosen uniformly in the region $0 \leqslant u \leqslant \widetilde{p}(\theta^{(\tau)})$, which defines a slice through the

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

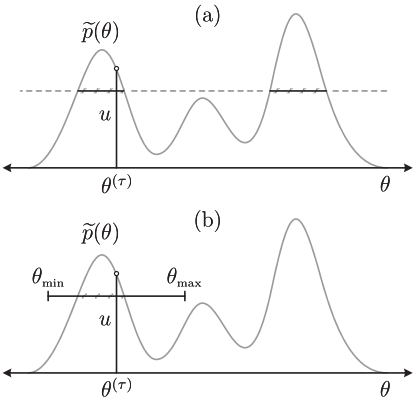


Figure 8.4 Slice sampling over a distribution $p(\theta)$ through: (a) finding the slice containing current sample $\theta^{(\tau)}$ with $\widetilde{p}(\theta) > u$; and (b) detecting the region of interest with two end points $\theta_{\min}$ and $\theta_{\max}$ so that we can uniformly draw a new sample $\theta^{(\tau+1)}$ within the slice. Adapted from Bishop (2006).

distribution, shown by solid horizontal lines. Because it is infeasible to sample directly from a slice, a new sample of θ is drawn from a region $\theta_{\min} \leqslant \theta \leqslant \theta_{\max}$, which contains the previous value $\theta^{(\tau)}$, as illustrated in Figure 8.4(b). We want the region to encompass as much of the slice as possible so as to allow large moves in θ space while having as little as possible of this region lying outside the slice, because this makes the sampling less efficient.

We start with a region of width w which contains current sample $\theta^{(\tau)}$ and then judge if both end points are within the slice. If either end point is within the slice, the region is extended in this direction by increments of value w until the end point falls outside the region. A candidate value $\theta^{\star}$ is then chosen uniformly from this region. If it lies inside the slice, it forms new sample $\theta^{(\tau+1)}$. If it lies outside the slice, the region is shrunk such that $\theta^{\star}$ forms an end point. A new candidate point is drawn uniformly from this reduced region until a value of θ is found that lies within the slice. When applying slice sampling to multivariate distributions, we repeatedly sample each variable in turn in the manner of Gibbs sampling based on a conditional distribution $p(\theta_i|\boldsymbol{\theta}^{-i})$.

8.2 Bayesian nonparametrics

The sampling methods mentioned in Section 8.1 are widely employed to infer the Bayesian nonparametrics (BNP) which are now seen as a new trend in speech and language processing areas where the data representation is a particular concern and is extensively studied. The BNP learning aims to deal with the issue that probabilistic methods are often not viewed as sufficiently expressive because of a long list of limitations and assumptions on probability distribution and fixed model complexity. It is attractive to pursue an expressive probabilistic representation with a less assumption-laden approach to inference. We would like to move beyond the simple

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

fixed-dimensional random variables, e.g., multinomial distributions, Gaussians, and other exponential family distributions, and to mimic the flexibility in probabilistic representations. BNP methods avoid the restrictive assumptions of parametric models by defining distributions on function spaces.

Therefore, the stochastic process, containing an indexed collection of random variables, provides the flexibility to define probability distributions on spaces of probability distributions. We relax the limitation of finite parameterizations and consider the models over infinite-dimensional objects such as trees, lists, and collections of sets. The expressive data structures could be explored by computationally efficient reasoning and learning through the so-called *combinatorial stochastic processes* (Pitman 2006). BNP learning is accordingly developed from a Bayesian perspective by upgrading the priors in classic Bayesian analysis from parametric distributions to stochastic processes. Prior distributions are then replaced by the *prior process* in BNP inference. The flexible Bayesian learning and representation is conducted from the prior stochastic process to the posterior stochastic process. BNP involves the combinatorics of sums and products over prior and posterior stochastic processes and automatically learned model structure from the observed data. The model selection problem could be tackled by BNP learning.

8.2.1 Modeling via exchangeability

Some of the foundation of Bayesian inference is addressed in this section. The concept of *exchangeability* is critical in motivating BNP learning. Consider a probabilistic model with an infinite sequence of random factors or parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \cdots\}$ to be inferred from observation data $\mathbf{x} = \{x_1, x_2, \cdots\}$, which could be either continuous such as the *speech features* $\mathbf{O}$ or discrete such as the *word labels* W. We say that such a sequence is *infinitely exchangeable* if the joint probability distribution of any finite subset of those random variables is invariant to permutation. For any N, we have

$$p(\theta_1, \theta_2, \cdots, \theta_N) = p(\theta_{\pi(1)}, \theta_{\pi(2)}, \cdots, \theta_{\pi(N)}), \tag{8.23}$$

where π denotes a permutation. The assumption of *exchangeability* is weaker than that of *independence* among random variables. This assumption often better describes the data we encounter in realization of stochastic processes for BNP learning. De Finetti's theorem states that the infinite sequence is exchangeable if and only if for any N random variables the sequence has the following property (Bernardo & Smith 2009):

$$p(\theta_1, \theta_2, \cdots, \theta_N) = \int \prod_{i=1}^{N} p(\theta_i|G) dP(G). \tag{8.24}$$

There exists an underlying random measure G and a distribution function P such that random variables θ_i are conditionally independent given G, which is not restricted to be a finite-dimensional object.

The Pólya urn model is a probability model for sequentially labeling the balls in an urn. Consider an empty urn and a countably infinite collection of colors. Randomly pick a color according to some fixed distribution G_0 and place a ball having the same color in

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

the urn. For all subsequent balls, either choose a ball from the urn uniformly and return that ball to the urn with another ball of the same color, or choose a new color from G_0 and place a ball of that color k in the urn. We express this process mathematically by

$$p(\theta_i = k|\theta_1, \cdots, \theta_{i-1}) \propto \begin{cases} c_k & \text{if } \theta_j = k \text{ for some } j \in \{1, \cdots, i-1\}, \\ \alpha_0 & \text{otherwise,} \end{cases} \tag{8.25}$$

where $\alpha_0 > 0$ is a parameter of the process and c_k denotes the number of balls of color k. Even though we define the model by considering a particular ordering of the balls, the resulting distribution is independent of the order. It can be proved that the joint distribution $p(\theta_1, \theta_2, \cdots, \theta_N)$ is written as a product of conditionals given in Eq. (8.25) and the resulting expression is independent of the order of N random variables. The exchangeability in the Pólya urn model is confirmed. Because of this property, by De Finetti's theorem, the existence of an underlying probability measure G renders the ball colors conditionally independent. This random measure corresponds to a stochastic process known as the *Dirichlet process*, which is introduced in Section 8.2.2.

The property of exchangeability is essential for an MCMC inference procedure. Let us consider the joint distribution of $\boldsymbol{\theta}$ and $\mathbf{x}$ given by

$$p(\boldsymbol{\theta}, \mathbf{x}) = p(\theta_1, \theta_2, \cdots, \theta_N) \prod_{i=1}^{N} p(x_i|\theta_i), \tag{8.26}$$

which is viewed as the product of a prior in the first term and a likelihood function in the second term. The first term $p(\theta_1, \theta_2, \cdots, \theta_N)$ is modeled by the Pólya urn marginal distributions. In particular, our goal is to sample $\boldsymbol{\theta}$ from observation data $\mathbf{x}$ based on the Gibbs sampling. The problem is to sample a particular component θ_i while all of the other components are fixed. Because the joint distribution of $\{\theta_1, \cdots, \theta_N\}$ is invariant to permutation, we can freely permute the vector to move θ_i to the end of the list. The prior probability of the last component given all of the preceding components is given by the urn model as given in Eq. (8.25). We multiply each of the distributions by the likelihood function $p(x_i|\theta_i)$ and integrate with respect to θ_i. We assume that the prior measure G_0 and the likelihood function are conjugate so that the integral can be done in closed form. For each component, the derived result is the conditional distribution of θ_i given the other components and given x_i. This is done for different components $\{\theta_1, \cdots, \theta_N\}$ and the process iterates. This link between exchangeability and an efficient inference algorithm is important for BNP learning.

In addition, it is crucial to realize the Pólya urn model for Bayesian speech and language processing over a speech and text corpus. The ball means a word w_i or a feature vector $\mathbf{o}_t$ in the corpus and the ball color indicates the cluster label of this word or feature vector. This Pólya urn model defines a distribution of acoustic features or word labels which is not fixed in dimensionality and can be used to induce a distribution on partitions or clusterings. The distribution on partitions is known as the *Chinese restaurant process* (Aldous 1985), which is addressed in Section 8.2.4. The Chinese restaurant process and the Pólya urn model are viewed as the essentials of the BNP model for clustering where the random partition provides a prior on clusterings and the color associated with a cell can be represented by a distribution associated with a given cluster.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

8.2.2 Dirichlet process

The *Dirichlet process* (DP) (Ferguson 1973) plays a crucial role in BNP learning. It is a stochastic process for a random probability measure G over a measurable space Ω,

$$G \sim \text{DP}(\alpha_0, G_0), \tag{8.27}$$

such that, for any finite measurable partition $(A_1, A_2, \cdots, A_r)$, the random vector $(G(A_1, \cdots, G(A_r))$ is distributed as a finite-dimensional Dirichlet distribution with parameters $(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_r))$, i.e.

$$(G(A_1, \cdots, G(A_r)) \sim \text{Dir}(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_r)), \tag{8.28}$$

with two parameters, a positive scaling parameter or *concentration parameter*, $\alpha_0 > 0$, and a *base probability measure*, $G_0 \in \Omega$. This process is a measure of measures over space Ω where $G(\Omega) = 1$. To realize the Dirichlet process in Eq. (8.27), an infinite sequence of points $\{\phi_k\}$ is drawn independently from the base probability measure G_0 so that the probability measure of the process is established by

$$G = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k} \tag{8.29}$$

with probability 1 (Sethuraman 1994). In Eq. (8.29), δ_{ϕ_k} is an *atom* or a unit mass at the point ϕ_k and $\{\beta_k\}$ are the random weights which depend on the concentration parameter α_0. Note that the Dirichlet process G is random in two ways. One random process is for weights β_k while the other is for locations ϕ_k. We can say that Dirichlet process G in Eq. (8.29) has the *Dirichlet marginals* as given in Eq. (8.28). According to De Finetti's theorem, the DP is seen as the De Finetti mixture distribution underlying the Pólya urn model as addressed in Section 8.2.1. DP can be presented from the perspectives of the stick-breaking construction, the Pólya urn scheme, and a limit of finite mixture models which is detailed in Sections 8.2.3, 8.2.4, and 8.2.5 respectively.

8.2.3 DP: Stick-breaking construction

The stick-breaking construction for DP was presented by Sethuraman (1994). In general, the DP and the stick-breaking process (SBP) are essential tools in BNP. Consider the stick-breaking weights $\{\beta_k\}_{k=1}^{\infty}$ on a countably infinite set. We want to find a distribution of the non-negative mixture weights $\beta_1, \beta_2, \cdots$ having the property

$$\sum_{k=1}^{\infty} \beta_k = 1. \tag{8.30}$$

One solution to this problem is provided by a procedure known as "stick-breaking." Considering a unit-length stick, we independently draw a proportion β_k' in the kth break from a Beta distribution with a concentration parameter α_0:

$$\beta_k' | \alpha_0 \sim \text{Beta}(1, \alpha_0) \quad k = 1, 2, \cdots. \tag{8.31}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

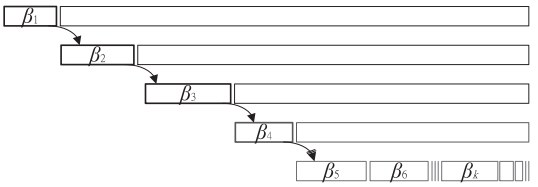


Figure 8.5 The stick-breaking process.

Each break decides a proportion β'_k while the proportion of the remaining stick is $1-\beta'_k$. The mixture weight of the first component is $\beta_1 = \beta'_1$ and that of the kth component is determined by

$$\beta_k = \beta'_k \prod_{l=1}^{k-1}(1-\beta'_l) \quad k = 1, 2, \cdots . \tag{8.32}$$

Under this SBP, it is straightforward to show the property of mixture weights that sum up to one with probability one. Figure 8.5 illustrates an infinite sequence of segments β_k from SBP. Let us define an infinite sequence of independent random variables $\{\beta_k\}_{k=1}^{\infty}$ and $\{\phi_k\}_{k=1}^{\infty}$ where

$$\phi_k | G_0 \sim G_0. \tag{8.33}$$

Then, the probability measure G defined in Eq. (8.29) can be shown to be the measure distributed according to DP(α_0, G_0) (Sethuraman 1994). We may interpret the sequence $\boldsymbol{\beta} = \{\beta_k\}_{k=1}^{\infty}$ as a random probability measure on the positive integers. This measure is formally denoted by the GEM distribution (Pitman 2002)

$$\boldsymbol{\beta} \sim \text{GEM}(\alpha_0). \tag{8.34}$$

8.2.4 DP: Chinese restaurant process

The second perspective on the DP is provided by the Pólya urn scheme (Blackwell & MacQueen 1973) showing that the draws from DP are both discrete and exhibit a clustering property. Let $\theta_1, \theta_2, \cdots$ be a sequence of independently and identically distributed (iid) random factors or parameters drawn from G for individual observations $x_1, x_2, \cdots$ under some distribution function. The probability model over the infinite sequence is written as

$$\begin{aligned} &\theta_i | G \sim G \\ &x_i | \theta_i \sim p(x_i | \theta_i) \qquad \text{for each } i. \end{aligned} \tag{8.35}$$

The factors $\theta_1, \theta_2, \cdots$ are conditionally independent given G, and hence are exchangeable. Consider the successive conditional distributions of θ_i of x_i given the previous factors $\theta_1, \cdots, \theta_{i-1}$ of observations $x_1, \cdots, x_{i-1}$, where G has been integrated out. It was shown that these conditional distributions have the following form (Blackwell & MacQueen 1973):

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

$$\theta_i|\theta_1,\cdots,\theta_{i-1},\alpha_0,G_0 \sim \sum_{l=1}^{i-1}\frac{1}{i-1+\alpha_0}\delta_{\theta_l}+\frac{\alpha_0}{i-1+\alpha_0}G_0. \tag{8.36}$$

We can interpret the conditional distributions as a simple urn model where a ball of a distinct color is associated with each atom δ_{θ_l}. The balls are drawn equiprobably or uniformly. As seen in the second term of Eq. (8.36), a new atom is created by drawing from G_0 with probability proportional to α_0. A ball of new color is added to the urn. Equation (8.36) means that θ_i has a positive probability of being equal to one of the previous draws $\theta_1,\cdots,\theta_{i-1}$. This process results in a reinforcement effect: the more often a point is drawn, the more probable it is to be drawn in the future. To make the clustering more explicitly, we introduce a new set of variables that represent distinct values of atoms. Let $\phi_1,\cdots,\phi_K$ denote the distinct values taken from previous factors $\theta_1,\cdots,\theta_{i-1}$ and c_k be the number of customers or values $\theta_{i'}$ that are sitting at or are associated with ϕ_k of table or cluster k. Equation (8.36) is re-expressed by

$$\theta_i|\theta_1,\cdots,\theta_{i-1},\alpha_0,G_0 \sim \sum_{k=1}^{K}\frac{c_k}{i-1+\alpha_0}\delta_{\phi_k}+\frac{\alpha_0}{i-1+\alpha_0}G_0. \tag{8.37}$$

From this re-expression, we find that the Pólya urn scheme produces a distribution on partitions which is closely related to the one using a different metaphor called the Chinese restaurant process (CRP) (Aldous 1985). The metaphor of CRP is addressed as follows. Consider a Chinese restaurant with an unbounded number of tables. Each θ_i corresponds to a customer x_i who enters the restaurant. The distinct values ϕ_k correspond to the tables at which the customers sit. The ith customer sits at the table indexed by ϕ_k with probability proportional to the number of customers c_k who are already seated there (in this case we set $\theta_i=\phi_k$), or sits at a new table with probability proportional to α_0 (in this case, we increment K, draw $\phi_K\sim G_0$, and set $\theta_i=\phi_K$). Figure 8.6 shows the scenario of the CRP where the current customer θ_{11} either chooses an occupied table from $\{\phi_1,\cdots,\phi_4\}$ or a new table ϕ_{new} according to

$$\begin{aligned} p(\text{occupied table } k|\text{previous customers}) &= \frac{c_k}{i-1+\alpha_0},\\ p(\text{next unoccupied table}|\text{previous customers}) &= \frac{\alpha_0}{i-1+\alpha_0}. \end{aligned} \tag{8.38}$$

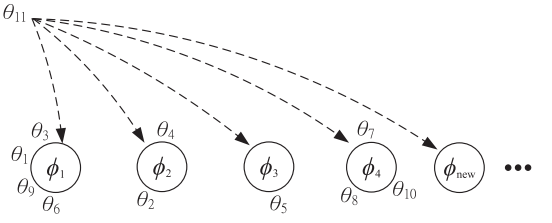


Figure 8.6 The Chinese restaurant process.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

The CRP probability model in Eq. (8.38) is closely related to the Pólya urn model in Eq. (8.25).

8.2.5 Dirichlet process mixture model

One of the most important applications of the DP is to explore the nonparametric prior over the parameters of a mixture model. The resulting model is referred to as the *DP mixture model* (Antoniak 1974). Consider the probability model of an infinite sequence of observations $x_1, x_2, \cdots$ in Eq. (8.35), with graphical representation as in Figure 8.7(a). The probability measure G can be represented by using a stick-breaking construction. The factors θ_i take on values ϕ_k with probability β_k. Here, an indicator value z_i is introduced to reveal the positive integer value or cluster index for factor θ_i, and it is distributed according to $\boldsymbol{\beta}$. A DP mixture model can be represented by the following conditional distributions:

$$\begin{aligned} \boldsymbol{\beta}|\alpha_0 &\sim \text{GEM}(\alpha_0), \\ z_i|\boldsymbol{\beta} &\sim \boldsymbol{\beta}, \\ \phi_k|G_0 &\sim G_0, \\ x_i|z_i, \{\phi_k\}_{k=1}^{\infty} &\sim p(x_i|\phi_{z_i}). \end{aligned} \tag{8.39}$$

Therefore, we have the mixture model $G = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}$ and $\theta_i = \phi_{z_i}$.

Alternatively, the DP mixture model can be derived as the limit of a sequence of finite mixture models where the number of mixture components is taken to infinity (Neal 1992). This limiting process provides the third perspective on DP. Suppose that we have K mixture components. Let $\boldsymbol{\beta} = \{\beta_1, \cdots, \beta_K\}$ denote the mixture weights. In the limit $K \to \infty$, the vectors $\boldsymbol{\beta}$ are closely related and are equivalent up to a random size-biased permutation. We use a *Dirichlet prior* on $\boldsymbol{\beta}$ with symmetric parameters $(\alpha_0/L, \cdots, \alpha_0/L)$. We thus have the following model:

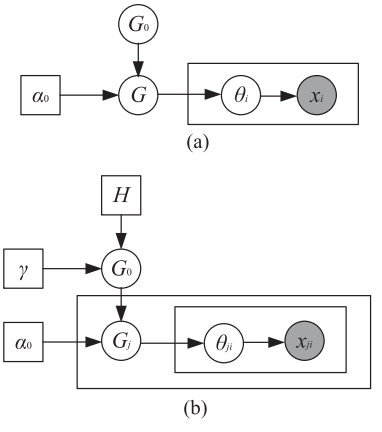


Figure 8.7 (a) Dirichlet process mixture model; (b) hierarchical Dirichlet process mixture model.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

$$\begin{aligned}\boldsymbol{\beta}|\alpha_0 &\sim \text{Dir}(\alpha_0/L, \cdots, \alpha_0/L),\\ z_i|\boldsymbol{\beta} &\sim \boldsymbol{\beta},\\ \phi_k|G_0 &\sim G_0,\\ x_i|z_i, \{\phi_k\}_{k=1}^K &\sim p(x_i|\phi_{z_i}).\end{aligned} \tag{8.40}$$

The corresponding mixture model $G^K = \sum_{k=1}^K \beta_k \delta_{\phi_k}$ was shown to have the property

$$\int f(\theta) dG^K(\theta) \longrightarrow \int f(\theta) dG(\theta), \tag{8.41}$$

as $K \to \infty$ for every measurable function $f(\cdot)$. The marginal distribution of the observations $x_1, \cdots, x_n$ approaches that using a DP mixture model.

8.2.6 Hierarchical Dirichlet process

The spirit of the graphical model based on directed graphs, as mentioned in Section 2.2, is mainly from that of hierarchical Bayesian modeling, while the graphical model literature has focused almost exclusively on parametric hierarchies where each of the conditionals is a finite-dimensional distribution. Nevertheless, it is possible to build hierarchies in which the components are stochastic processes. We illustrate how to do this based on the Dirichlet process.

In particular, we are interested in finding solutions to the problems in which the observations are organized into groups, where the observations are assumed to be exchangeable both within each group and across groups. For example, in document representation, the words in each document in a text corpus are associated with the data in each group from a set of grouped data. We want to discover the structures of words from a set of training documents. It is important to find the clusterings of word labels for data representation. Let j index the groups or documents and i index the observations or words within each group. We assume that $x_{j1}, x_{j2}, \cdots$ are exchangeable within each group j and also between groups. The group data $\mathbf{x}_j = \{x_{j1}, x_{j2}, \cdots\}$ in an infinite set of groups $\mathbf{x}_1, \mathbf{x}_2, \cdots$ are exchangeable.

Assuming that each observation is drawn independently from a mixture model, each observation x_{ji} is associated with a mixture component k. Let θ_{ji} denote a parameter or factor specifying the mixture component ϕ_k associated with the observation x_{ji}. The factors θ_{ji} are not generally distinct. Let $p(x_{ji}|\theta_{ji})$ denote the distribution of observation x_{ji} given the factor θ_{ji}. Let G_j denote a distribution for the factors $\boldsymbol{\theta}_j = \{\theta_{j1}, \theta_{j2}, \cdots\}$ associated with group j. Due to exchangeability, the factors are conditionally independent given G_j. The probability model for this stochastic process is expressed by

$$\begin{aligned}\theta_{ji}|G_j &\sim G_j,\\ x_{ji}|\theta_{ji} &\sim p(x_{ji}|\theta_{ji}) \quad \text{for each } j \text{ and } i.\end{aligned} \tag{8.42}$$

The hierarchical Dirichlet process (HDP) (Teh *et al*. 2006) was proposed to conduct BNP learning of grouped data, in which each group is associated with a mixture model and we wish to link these mixture models. An HDP is a nonparametric prior distribution

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

over a set of random probability measures. The process defines a set of probability measures G_j, one for each group j, and a global probability measure G_0 shared for different groups. The nonparametric priors in HDP are given by the following hierarchy:

$$\begin{aligned} G_0|\gamma, H &\sim \mathrm{DP}(\gamma, H), \\ G_j|\alpha_0, G_0 &\sim \mathrm{DP}(\alpha_0, G_0) \quad \text{for each } j. \end{aligned} \tag{8.43}$$

The prior random measure of jth group G_j is a DP with a shared base measure G_0 for grouped data which is itself drawn from the DP with a base measure H. Here, the hyperparameters γ and α are the concentration parameters and H provides the prior distribution for the factors θ_{ji}. Basically, the distribution G_0 varies around the prior H with variations governed by γ while the distribution G_j over the factors in the jth group deviates from G_0 with variations controlled by α_0. Analogous to the DP mixture model mentioned in Section 8.2.5, HDP is feasible to construct the HDP mixture model as shown graphically in Figure 8.7(b). The HDP mixture model is expressed by:

$$\begin{aligned} G_0|\gamma, H &\sim \mathrm{DP}(\gamma, H), \\ G_j|\alpha_0, G_0 &\sim \mathrm{DP}(\alpha_0, G_0) \quad \text{for each } j, \\ \theta_{ji}|G_j &\sim G_j, \\ x_{ji}|\theta_{ji} &\sim p(x_{ji}|\theta_{ji}) \quad \text{for each } j \text{ and } i. \end{aligned} \tag{8.44}$$

We can see that this model achieves the goal of sharing clusters across groups by assigning the same parameters or factors to those observations. That is, if $\theta_{ji} = \theta_{ji'}$, the observations x_{ji} and $x_{ji'}$ belong to the same cluster. The equality of factors is possible because both θ_{ji} and $\theta_{ji'}$ are possibly drawn from G_j, which is a discrete measure over a measurable partition $(A_1, A_2, \cdots)$. Since G_j from different groups shares the atoms from G_0, the observations in different groups j can be assigned to the same cluster. In what follows, we present the realization of the HDP from the perspectives of a stick-breaking construction and a Chinese restaurant process.

8.2.7 HDP: Stick-breaking construction

Using a stick-breaking construction, the global measure G_0 and the individual measure G_j in HDP are expressed by the mixture models with the shared atoms $\{\phi_k\}_{k=1}^{\infty}$ but different weights $\boldsymbol{\beta} = \{\beta_k\}_{k=1}^{\infty}$ and $\boldsymbol{\pi}_j = \{\pi_{jk}\}_{k=1}^{\infty}$, respectively, as given by:

$$\begin{aligned} G_0 &= \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}, \\ G_j &= \sum_{k=1}^{\infty} \pi_{jk} \delta_{\phi_k}, \end{aligned} \tag{8.45}$$

where atom ϕ_k is drawn from base measure H and weights $\boldsymbol{\beta}$ are drawn from the GEM distribution $\boldsymbol{\beta} \sim \mathrm{GEM}(\gamma)$. Note that the weights $\boldsymbol{\pi}_j$ are independent given $\boldsymbol{\beta}$ because the G_js are independent given G_0.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

Let $(A_1, \cdots, A_r)$ denote a measurable partition and $K_r = k : \phi_k \in A_l$ for $l = 1, \cdots, r$. Here, $(K_1, \cdots, K_r)$ is a finite partition of the positive integers. For each group j, we have

$$\begin{aligned}
&(G_j(A_i), \cdots, G_j(A_r)) \\
&\quad \sim \text{Dir}(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_r)) \\
&\Rightarrow \left(\sum_{k \in K_1} \pi_{jk}, \cdots, \sum_{k \in K_r} \pi_{jk} \right) \\
&\quad \sim \text{Dir} \left(\alpha_0 \sum_{k \in K_1} \beta_k, \cdots, \alpha_0 \sum_{k \in K_r} \beta_k \right) \\
&\Rightarrow \boldsymbol{\pi}_j \sim \text{DP}(\alpha_0, \boldsymbol{\beta}).
\end{aligned} \tag{8.46}$$

Hence, each $\boldsymbol{\pi}_j$ is independently distributed according to $\text{DP}(\alpha_0, \boldsymbol{\beta})$. The HDP mixture model is then represented by:

$$\begin{aligned}
\boldsymbol{\beta}|\gamma &\sim \text{GEM}(\gamma), \\
\boldsymbol{\pi}_j|\alpha_0, \boldsymbol{\beta} &\sim \text{DP}(\alpha_0, \boldsymbol{\beta}), \\
z_{ji}|\boldsymbol{\pi}_j &\sim \boldsymbol{\pi}_j, \\
\phi_k|H &\sim H, \\
x_{ji}|z_{ji}, \{\phi_k\}_{k=1}^{\infty} &\sim p(x_{ji}|\phi_{z_{ji}}).
\end{aligned} \tag{8.47}$$

We may further show the explicit relationship between the elements of $\boldsymbol{\beta}$ and $\boldsymbol{\pi}_j$. The stick-breaking construction for $\text{DP}(\gamma, H)$ defines the variables β_k as

$$\begin{aligned}
\beta'_k &\sim \text{Beta}(1, \gamma), \\
\beta_k &= \beta'_k \prod_{l=1}^{k-1} (1 - \beta'_l).
\end{aligned} \tag{8.48}$$

Also, the stick-breaking construction for a probability measure $\boldsymbol{\pi}_j \sim \text{DP}(\alpha_0, \boldsymbol{\beta})$ of group j is performed by

$$\begin{aligned}
\pi'_{jk} &\sim \text{Beta} \left(\alpha_0 \beta_k, \alpha_0 \sum_{l=k+1}^{\infty} \beta_l \right) \\
&= \text{Beta} \left(\alpha_0 \beta_k, \alpha_0 \left(1 - \sum_{l=1}^{k} \beta_l \right) \right), \\
\pi_{jk} &= \pi'_{jk} \prod_{l=1}^{k-1} (1 - \pi'_{jl}).
\end{aligned} \tag{8.49}$$

This completes the stick-breaking construction for the HDP.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

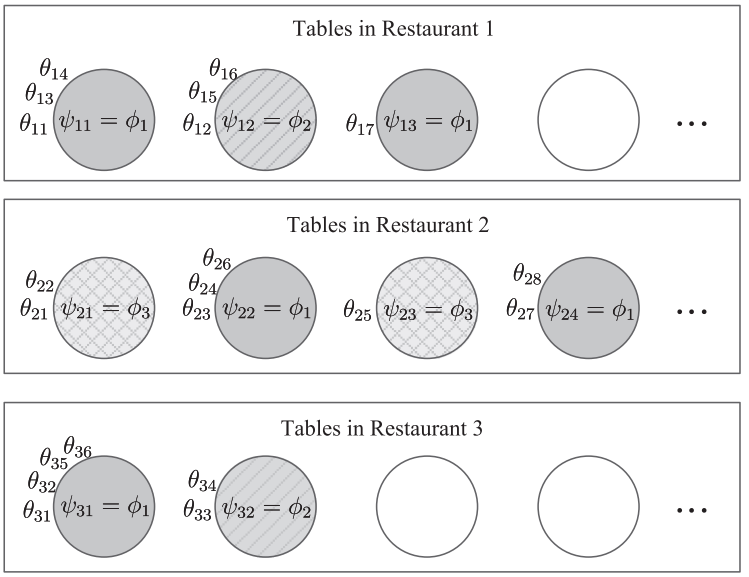


Figure 8.8 A Chinese restaurant franchise containing three restaurants (rectangles) with an infinite number of tables (circles) and dishes $\{\phi_k\}_{k=1}^{\infty}$. The term θ_{ji} denotes the ith customer in restaurant j, and ψ_{jt} is the indicator of a dish on the tth table of restaurant j. Different dishes ϕ_1, ϕ_2, and ϕ_3 are marked by different shading patterns.

8.2.8 HDP: Chinese restaurant franchise

The Chinese restaurant process for a DP is further extended to the *Chinese restaurant franchise* for an HDP, which allows for multiple restaurants sharing a set of dishes, as depicted in Figure 8.8. The metaphor is as follows. There is a restaurant franchise which shares the menu across restaurants. At each table of each restaurant, one dish is ordered from the menu by the first customer who sits there. This dish is shared among all customers sitting at that table. Different tables in different restaurants can serve the same dish. In this scenario, the restaurants correspond to the groups while the customers correspond to the factors θ_{ji} of observations x_{ji}. Let $\phi_1, \cdots, \phi_K$ denote the dishes in a global menu which are drawn from H. We introduce the variable, ψ_{jt}, that indicates the dish served at table t in restaurant j. In this restaurant franchise, we first consider the conditional distribution for a customer θ_{ji}, given the previous customers $\theta_{j1}, \cdots, \theta_{j,i-1}$ in restaurant j, and G_0, where the DP G_j is integrated out. From Eq. (8.37), we obtain

$$\theta_{ji}|\theta_{j1}, \cdots, \theta_{j,i-1}, \alpha_0, G_0 \sim \sum_{t=1}^{m_{j\cdot}} \frac{c_{jt\cdot}}{i-1+\alpha_0}\delta_{\psi_{jt}} + \frac{\alpha_0}{i-1+\alpha_0}G_0. \tag{8.50}$$

The notation c_{jtk} denotes the number of customers in restaurant j at table t eating dish k. Marginal count is represented by dot. Thus, $c_{jt\cdot}$ denotes the number of customers in restaurant j at table t. The notation m_{jk} means the number of tables in restaurant j serving dish k. Thus, $m_{j\cdot}$ represents the number of tables in restaurant j. This conditional is a

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

mixture or a draw, which can be obtained by drawing from the terms on the right-hand-side of Eq. (8.50) with the probabilities given by the corresponding mixture weights for the occupied tables and an unoccupied table. If a term in the first summation is chosen, we increment $c_{jt\cdot}$ and set $\theta_{ji} = \psi_{jt}$. If the second term is chosen, we increment $m_{j\cdot}$, draw $\psi_{jm_{j\cdot}} \sim G_0$ and set $\theta_{ji} = \psi_{jm_{j\cdot}}$.

Next, we proceed to integrate out G_0, which is a DP, and apply Eq. (8.37) again to find the conditional distribution of a factor ψ_{jt} given the previous factors in the different restaurants:

$$\psi_{jt}|\psi_{11}, \psi_{12}, \cdots, \psi_{21}, \cdots, \psi_{j,t-1}, \gamma, H \sim \sum_{k=1}^{K} \frac{m_{\cdot k}}{m_{\cdot\cdot} + \gamma} \delta_{\phi_k} + \frac{\gamma}{m_{\cdot\cdot} + \gamma} H. \tag{8.51}$$

If we draw ψ_{jt} by choosing the term in the summation on the right-hand-side of Eq. (8.51), we set $\psi_{jt} = \phi_k$. If we choose the second term, we increment K, draw a new $\phi_K \sim H$ and set $\psi_{jk} = \phi_K$. It is meaningful that the mixture probability of the first term is proportional to $m_{\cdot k}$, which represents the number of tables serving dish k, while the probability of the second term is proportional to the concentration parameter γ of DP G_0. This completes the HDP implementation based on the Chinese restaurant franchise. To obtain samples θ_{ji} for each j and i, we first draw θ_{ji} from Eq. (8.50). If a new sample from G_0 is needed, we use Eq. (8.51) to obtain a new sample ψ_{jt} and set $\theta_{ji} = \psi_{jt}$.

Moreover, the HDP can be derived from the perspective of infinite limit over the finite mixture models. This is done by considering the collection of finite mixture models where L-dimensional $\boldsymbol{\beta}$ and $\boldsymbol{\pi}_j$ are used as the global and group-dependent mixing probabilities, respectively. The HDP finite mixture model is accordingly expressed by

$$\begin{aligned} \boldsymbol{\beta}|\gamma &\sim \mathrm{Dir}(\gamma/L, \cdots, \gamma/L), \\ \boldsymbol{\pi}_j|\alpha_0, \boldsymbol{\beta} &\sim \mathrm{Dir}(\alpha_0 \boldsymbol{\beta}), \\ z_{ji}\boldsymbol{\pi}_j &\sim \boldsymbol{\pi}_j, \\ \phi_k|H &\sim H, \\ x_{ji}|z_{ji}, \{\phi_k\}_{k=1}^{L} &\sim p(x_{ji}|\phi_{z_{ji}}). \end{aligned} \tag{8.52}$$

It can be shown that the limit of this model as $L \to \infty$ approaches the HDP mixture model. The parametric hierarchical prior for $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ in Eq. (8.52) is also seen as the hierarchical Dirichlet model, which has been applied for language modeling (MacKay & Peto 1995), as addressed in Section 5.3.

8.2.9 MCMC inference by Chinese restaurant franchise

The MCMC sampling schemes have been developed for inference of the HDP mixture model. A straightforward Gibbs sampler based on the Chinese restaurant franchise can be implemented. To do so, the posterior probabilities for drawing tables $\mathbf{t} = \{t_{ji}\}$ and dishes $\mathbf{k} = \{k_{jt}\}$ should be determined. Let t_{ji} be the index of the factor ψ_{jt} associated with θ_{ji}, and let k_{jt} be the index of ϕ_k associated with ψ_{jt}. In the Chinese restaurant franchise, the customer i in restaurant j sits at table t_{ji}, whereas table t in restaurant j

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

serves dish k_{jt}. Let $\mathbf{x} = \{x_{ji}\}$, $\mathbf{x}_{jt} = \{x_{ji}, \text{all } i \text{ with } t_{ji} = t\}$, $\mathbf{z}_{z_{ji}}$, $\mathbf{m} = m_{jk}$ and $\boldsymbol{\phi} = \{\phi_1, \cdots, \phi_K\}$. Notations $\mathbf{k}^{-jt} = \mathbf{k} \backslash k_{jt}$ and $\mathbf{t}^{-ji} = \mathbf{t} \backslash t_{ji}$ denote the vectors $\mathbf{k}$ and $\mathbf{t}$ with the exception of components k_{jt} and t_{ji}, respectively, and $c_{jt\cdot}^{-ji}$ denotes the number of customers in restaurant j who enjoy the dish ψ_{jt}, but leaving out customer x_{ji}. Similarly, we have $\mathbf{x}^{-ji} = \mathbf{x} \backslash x_{ji}$ and $\mathbf{x}^{-jt} = \mathbf{x} \backslash \mathbf{x}_{jt}$. The concentration parameters γ and α_0 of DPs are assumed to be fixed. When implementing HDP using MCMC sampling, we need to calculate the conditional distribution of x_{ji} under mixture component k given all data samples except x_{ji},

$$p(x_{ji}|\mathbf{x}^{-ji}, k) = \frac{\int p(x_{ji}|\phi_k) \prod_{j'i' \neq ji, z_{j'i'}=k} p(x_{j'i'}|\phi_k) h(\phi_k) d\phi_k}{\int \prod_{j'i' \neq ji, z_{j'i'}=k} p(x_{j'i'}|\phi_k) h(\phi_k) d\phi_k}. \tag{8.53}$$

Here, $h(\phi_k)$ is a prior distribution from H and is conjugate to the likelihood $p(x|\phi_k)$. We can integrate out the mixture component parameter ϕ_k in closed form. Similarly to Eq. (8.53), the conditional distribution $p(\mathbf{x}_{jt}|\mathbf{x}^{-jt}, k)$ can be calculated by using $\mathbf{x}_{jt}$, the customers at restaurant j sitting at table t. In what follows, we derive the posterior probabilities for sampling tables t_{ji} and dishes k_{jt} which are then used to reconstruct θ_{ji}s and ψ_{jt}s based on the ϕ_ks. The property of exchangeability is employed in Eqs. (8.50) and (8.51) for θ_{ji}s and ψ_{jt}s, or equivalently for t_{ji}s and k_{jt}s, respectively.

Sampling $\mathbf{t}$: We want to calculate the conditional distribution of t_{ji} given the rest of variables $\mathbf{t}^{-ji}$. This conditional posterior for t_{ji} is obtained by combining the conditional prior for t_{ji} with the likelihood of generating x_{ji}. Basically, the likelihood function due to x_{ji} given previously occupied table t is expressed by $p(x_{ji}|\mathbf{x}^{-ji}, k)$, as shown in Eq. (8.53). The likelihood function for a new table $t_{ji} = t_{\text{new}}$ can be calculated by integrating out all possible values of $k_{jt_{\text{new}}}$ using Eq. (8.51):

$$\begin{aligned} &p(x_{ji}|\mathbf{t}^{-ji}, t_{ji} = t_{\text{new}}, \mathbf{k}) \\ &\quad = \sum_{k=1}^{K} \frac{m_{\cdot k}}{m_{\cdot\cdot} + \gamma} p(x_{ji}|\mathbf{x}^{-ji}, k) + \frac{\gamma}{m_{\cdot\cdot} + \gamma} p(x_{ji}|\mathbf{x}^{-ji}, k_{\text{new}}), \end{aligned} \tag{8.54}$$

where $p(x_{ji}|\mathbf{x}^{-ji}, k_{\text{new}})$ is a prior density of x_{ji} given by

$$p(x_{ji}) = \int p(x_{ji}|\phi) h(\phi) d\phi. \tag{8.55}$$

According to Eq. (8.50), the conditional prior probability of taking a previously occupied table $t_{ji} = t$ is proportional to $c_{jt\cdot}^{-ji}$, while the probability of taking a new table $t = t_{\text{new}} = m_{j\cdot} + 1$ is proportional to α_0. Thus, the conditional posterior probability for sampling table t_{ji} is derived from

$$\begin{aligned} &p(t_{ji} = t|\mathbf{t}^{-ji}, \mathbf{x}, \mathbf{k}) \\ &\quad \propto \begin{cases} c_{jt\cdot}^{-ji} \cdot p(x_{ji}|\mathbf{x}^{-ji}, k_{jt}) & \text{if } t \text{ previously occupied,} \\ \alpha_0 \cdot p(x_{ji}|\mathbf{t}^{-ji}, t_{ji} = t_{\text{new}}, \mathbf{k}) & \text{if } t = t_{\text{new}}. \end{cases} \end{aligned} \tag{8.56}$$

Sampling $\mathbf{k}$: When the sampled table is a new one, $t_{ji} = t_{\text{new}}$, we need to further order a new dish $k_{jt_{\text{new}}}$ from the global menu. This is done according to the conditional

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

posterior probability, which is combined from a conditional prior for k_{jt} and a likelihood function due to x_{ji}. The conditional posterior probability is then derived from

$$p(k_{jt_{\text{new}}} = k|\mathbf{t}, \mathbf{x}, \mathbf{k}^{-jt_{\text{new}}}) \propto \begin{cases} m_{\cdot k} \cdot p(x_{ji}|\mathbf{x}^{-ji}, k) & \text{if } k \text{ previously ordered,} \\ \gamma \cdot p(x_{ji}|\mathbf{x}^{-ji}, k_{\text{new}}) & \text{if } k = k_{\text{new}}. \end{cases} \tag{8.57}$$

Because changing k_{jt} actually changes the component membership of all customers $\mathbf{x}_{jt}$ at table t, we can calculate the conditional posterior probability of ordering dish $k_{jt} = k$ from

$$p(k_{jt} = k|\mathbf{t}, \mathbf{x}, \mathbf{k}^{-jt}) \propto \begin{cases} m_{\cdot k}^{-jt} \cdot p(\mathbf{x}_{jt}|\mathbf{x}^{-jt}, k) & \text{if } k \text{ previously ordered,} \\ \gamma \cdot p(\mathbf{x}_{jt}|\mathbf{x}^{-jt}, k_{\text{new}}) & \text{if } k = k_{\text{new}}. \end{cases} \tag{8.58}$$

where the likelihood function $p(\mathbf{x}_{jt}|\mathbf{x}^{-jt}, k)$ due to the customers $\mathbf{x}_{jt}$ at table t of restaurant j is used.

8.2.10 MCMC inference by direct assignment

In the first MCMC procedure based on the Chinese restaurant franchise representation, the observations are first assigned to some table t_{ji}, and the tables are then assigned to some mixture component k_{jt}. This indirect association with mixture components could be realized by directly assigning mixture components through a new variable z_{ji} which is the same as $k_{jt_{ji}}$. The variable z_{ji} indicates the index of a mixture component corresponding to the customer x_{ji}. Using this direct assignment, the tables are represented only in terms of the number of tables m_{jk}. A bookkeeping scheme is involved. We would like to instantiate and sample from G_0 by using the *factorized posterior* on G_0 across groups. To do so, an explicit construction for $G_0 \sim \text{DP}(\gamma, H)$ is expressed in the form of

$$\begin{aligned} \boldsymbol{\beta} = (\beta_1, \cdots, \beta_K, \beta_u) &\sim \text{Dir}(m_{\cdot 1}, \cdots, m_{\cdot K}, \gamma), \\ G_u &\sim \text{DP}(\gamma, H), \\ p(\phi_k|\mathbf{z}) &\propto h(\phi_k) \prod_{ji:z_{ji}=k} p(x_{ji}|\phi_k, z_{ji}), \\ G_0 &= \sum_{k=1}^{K} \beta_k \delta_{\phi_k} + \beta_u G_u, \end{aligned} \tag{8.59}$$

which can be also expressed as

$$G_0 \sim \text{DP}\left(\gamma + m_{\cdot\cdot}, \frac{\gamma H + \sum_{k=1}^{K} m_{\cdot k} \delta_{\phi_k}}{\gamma + m_{\cdot\cdot}}\right). \tag{8.60}$$

From the expressions in Eq. (8.59), we can see that the values $m_{\cdot k}$ and γ in the first MCMC inference based on the *Chinese restaurant franchise* are replaced by β_k and β_u in the second MCMC inference based on the respective *direct assignments*. In the MCMC

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

inference based on direct assignment, we need to sample the indicators of mixture components $\mathbf{z} = \{z_{ij}\}$ and the numbers of tables $\mathbf{m} = \{m_{jk}\}$, according to the corresponding posterior probabilities which are provided in what follows.

Sampling $\mathbf{z}$: The idea of sampling $\mathbf{z}$ is to group together the terms associated with each k in Eqs. (8.54) and (8.56) so as to calculate the conditional posterior probability:

$$p(z_{ji} = k|\mathbf{z}^{-ji}, \mathbf{x}, \mathbf{m}, \boldsymbol{\beta}) = \begin{cases} (c_{j\cdot k}^{-ji} + \alpha_0\beta_k)p(x_{ji}|\mathbf{x}^{-ji}, k) & \text{if } k \text{ previously occupied,} \\ \alpha_0\beta_u p(x_{ji}|\mathbf{x}^{-ji}, k_{\text{new}}) & \text{if } k = k_{\text{new}}. \end{cases} \tag{8.61}$$

Note that we have combined Eqs. (8.54) and (8.56) based on a new variable z_{ji} and have replaced $m_{\cdot k}$ with β_k and γ with β_u. To fulfil the sampling of $\mathbf{z}$ in Eq. (8.61), we have to further sample $\mathbf{m}$ and $\boldsymbol{\beta}$.

Sampling $\mathbf{m}$: According to the direct assignment of data items to mixture components $\mathbf{z}$, it is sufficient to sample $\mathbf{m}$ and $\boldsymbol{\beta}$ in place of $\mathbf{t}$ and $\mathbf{k}$. To find the conditional distribution of m_{jk}, we consider the conditional distribution of t_{ji} under the condition $k_{jt_{ji}} = z_{ji}$. From Eq. (8.50), the prior probability that data item x_{ji} is assigned to some table t such that $k_{jt} = k$ is

$$p(t_{ji} = t|k_{jt} = k, \mathbf{t}^{-ji}, \mathbf{k}, \boldsymbol{\beta}) \propto c_{jt\cdot}^{-ji}, \tag{8.62}$$

while the probability that is assigned to a new table under mixture component k is

$$p(t_{ji} = t_{\text{new}}|k_{jt_{\text{new}}} = k, \mathbf{t}^{-ji}, \mathbf{k}, \boldsymbol{\beta}) \propto \alpha_0\beta_k. \tag{8.63}$$

These probabilities in a Gibbs sampler have the equilibrium distribution, which is the prior probability over the assignment of $c_{j\cdot k}$ observations to components in a DP with concentration parameter $\alpha_0\beta_k$. The corresponding distribution over the number of mixture components is the desired conditional distribution of m_{jk} which is written as (Antoniak 1974, Teh *et al*. 2006):

$$p(m_{jk} = m|\mathbf{z}, \mathbf{m}^{-jk}, \boldsymbol{\beta}) = \frac{\Gamma(\alpha_0\beta_k)}{\Gamma(\alpha_0\beta_k + c_{j\cdot k})} s(c_{j\cdot k}, m)(\alpha_0\beta_k)^m. \tag{8.64}$$

Here, $s(c, m)$ denotes the unsigned Stirling numbers of the first kind, which are calculated from

$$s(c + 1, m) = s(c, m - 1) + cs(c, m), \tag{8.65}$$

with initial conditions $s(0, 0) = s(1, 1) = 1$, $s(c, 0) = 0$ for $c > 0$ and $s(c, m) = 0$ for $m > c$.

Sampling $\boldsymbol{\beta}$: Having the samples $\mathbf{m}$ and the fixed hyperparameter γ, the β_k parameters are sampled from a Dirichlet distribution,

$$(\beta_1, \cdots, \beta_K, \beta_u)|\mathbf{m}, \gamma \sim \text{Dir}(m_{\cdot 1}, \cdots, m_{\cdot K}, \gamma). \tag{8.66}$$

This completes the MCMC inference for the HDP, based on the scheme of direct assignment of mixture components. It was shown that MCMC inference with the scheme of direct assignment is better than the scheme using the Chinese restaurant franchise in terms of convergence speed.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

8.2.11 Relation of HDP to other methods

In general, HDP can be seen as a building block for a variety of speech and language processing applications. An instance of application is the latent Dirichlet allocation (LDA) model (Blei *et al*. 2003), where each entity is associated not with a single cluster but with a set of clusters. In LDA terminology, each document is associated with a set of topics. As described in Section 7.6, an LDA model is constructed as a Bayesian parametric model with a fixed number of topics. HDP relaxes the limitation of a finite dimension in latent topic space in LDA and builds the flexible topic model with infinite clusters or topics. Multiple DPs are used to capture the uncertainty regarding the number of mixture components. HDP is viewed as a BNP version of an LDA model. The topics or clusters ϕ_k for the jth document are drawn from the nonparametric prior G_j, while the measures G_j are drawn from a DP with a base measure G_0. This allows the same topics to appear in multiple documents.

There are some other ways to connect multiple DPs. One idea is based on the *nested Dirichlet process* (NDP) (Rodriguez, Dunson & Gelfand 2008), which was proposed to model a collection of dependent distributions by using random variables as atoms at the higher level and random distributions as atoms at the lower level. This combinatorial process borrows information across DPs while also allowing DPs to be clustered. The simultaneous multilevel clustering can be achieved by such a nested setting. NDP is characterized by

$$\begin{aligned} G_j &\sim Q \quad \text{for each } j, \\ Q &\sim \mathrm{DP}(\alpha_0, \mathrm{DP}(\gamma, H)). \end{aligned} \tag{8.67}$$

Using HDP, the distributions $\{G_j\}$ of different j share the same atoms but assign them with different weights. However, using NDP, these distributions may have completely different atoms and weights. By marginalizing over the DPs, the resulting urn model is closely related to the *nested Chinese restaurant process* (nCRP) (Blei, Griffiths & Jordan 2010), which is known as a tree model of Chinese restaurants. The scenario of nCRP is addressed as follows. A customer enters the tree at a root Chinese restaurant and sits at a table. This table points to another Chinese restaurant, where the customer goes to dine the next evening. The construction is done recursively. Thus, a given customer follows a path through the tree of restaurants. Through BNP inference of topic hierarchies, a hierarchical topic model is established (Blei, Griffiths, Jordan *et al*. 2004). Each document or customer chooses a tree path of topics while each word in the document is represented by a mixture model of hierarchical topics along this tree path. In the following sections, we address some BNP methods which have been successfully applied for building the speaker diarization system and for developing the solutions of acoustic and language models to speech recognition systems.

8.3 Gibbs sampling-based speaker clustering

This section describes an application of MCMC techniques for speech features. We focus on speaker clustering, as discussed in Section 6.6.2 based on a Bayesian

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

information criterion (BIC). Section 6.6.2 models a speaker cluster with a single Gaussian where we assume a stationary property for speech features within a segment. However, this is actually not correct since it has various temporal patterns based on linguistic variations, speaking styles, and noises. Therefore, we model these short-term variations with a GMM, while speaker cluster characteristics are modeled by a mixture of GMMs to consider the multi-scale properties of speech dynamics (Moraru, Meignier, Besacier *et al*. 2003, Valente & Wellekens 2004*b*, Wooters, Fung, Peskin *et al*. 2004, Meignier, Moraru, Fredouille *et al*. 2006).

An important aspect of this speech modeling technique is the consideration of the multi-scale property in dynamics within a probabilistic framework. For example, PLSA (in Section 3.7.3) and LDA (in Section 7.6) are successful approaches in terms of the multi-scale property. They deal accurately with two types of scales, namely, word-level and document-level scales (i and m in the complete data likelihood function of the latent topic model in Eq. (3.293)), based on a latent topic model (Hofmann 1999*a*). The approach discussed in this section is inspired by these successful approaches, and aims to apply a fully Bayesian treatment to the multi-scale properties of speech dynamics.

There have been several studies on Bayesian speech modeling, e.g., by using maximum a-posteriori (MAP) in Chapter 4 or variational Bayesian (VB) approaches in Chapter 7 for speech recognition (Gauvain & Lee 1994, Watanabe *et al*. 2004), speaker verification (Reynolds *et al*. 2000), and speaker clustering (Valente & Wellekens 2004*b*). While all of these approaches are based on the EM-type deterministic algorithm, this section focuses on another method of realizing fully Bayesian treatment, namely *sampling approaches* based on MCMC. The main advantage of the sampling approaches is that they can avoid local optimum problems in addition to providing other Bayesian advantages (mitigation of data sparseness problems and capability of model structure optimization). While their heavy computational cost could be a problem in realization, recent improvements in computational power and the development of theoretical and practical aspects related to the sampling approaches allow us to apply them to practical problems (e.g., Griffiths & Steyvers (2004), Goldwater & Griffiths (2007), Porteous, Newman, Ihler *et al*. (2008) in natural language processing). Therefore, the aim of this work is to apply a sampling approach to speech modeling considering the multi-scale properties of speech dynamics.

The following sections formulate the multi-scale GMM by utilizing a Gibbs sampling approach. In this section, for its educational value, we first describe Gibbs sampling for a standard GMM. Section 8.3.2 derives the marginal likelihood of the GMM, which is used for deriving GMM Gibbs samplers in Section 8.3.3. Section 8.3.4 provides the generative process and a graphical model of multi-scale GMM for speaker clustering. Based on the analytical results of GMM Gibbs sampling, Section 8.3.5 derives the marginal likelihood of the multi-scale GMM, which is used for deriving Gibbs samplers in Section 8.3.6.

8.3.1 Generative model

This section considers the two types of observation vector sequences. One is an utterance- (or segment-) level sequence and the other is a frame-level sequence. Then, a

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

D dimensional observation vector (e.g., MFCC) at frame t in utterance u is represented as $\mathbf{o}_{ut} \in \mathbb{R}^D$. A set of observation vectors in utterance u is represented as

$$\mathbf{O}_u \triangleq \{\mathbf{o}_{ut} \in \mathbb{R}^D | t = 1, \cdots, T_u\}. \tag{8.68}$$

T_u denotes the number of frames at an utterance u.

Next, we assume that the frame-level sequence is modeled by a GMM as usual, and the utterance-level sequence is modeled by a mixture of these GMMs. Two kinds of latent variables are involved in multi-scale GMM for each sequence: utterance-level latent variable z_u and frame-level latent variable v_{ut}. Utterance-level latent variables may represent emotion, topic, and speaking style as well as speakers, depending on the speech variation. The joint likelihood function of U observation vectors ($\mathbf{O} \triangleq \{\mathbf{O}_u | u = 1, \cdots, U\}$) with the latent variable sequences ($Z \triangleq \{z_u \in \{1, \cdots, S\} | u = 1, \cdots, U\}$ and $V \triangleq \{v_{ut} \in \{1, \cdots, K\} | t = 1, \cdots, T_u, u = 1, \cdots, U\}$) can be expressed as follows:

$$p(\mathbf{O}, Z, V|\Theta) = \prod_{u=1}^{U} h_{z_u} \prod_{t=1}^{T_u} w_{z_u v_{ut}} \mathcal{N}(\mathbf{o}_{ut} | \boldsymbol{\mu}_{z_u v_{ut}}, \mathbf{R}^{-1}_{z_u v_{ut}}), \tag{8.69}$$

where $h_s \in [0, 1]$ denotes the utterance-level mixture weight, and $w_{sk} \in [0, 1]$ denotes the frame-level mixture weight. The terms $\boldsymbol{\mu}_{sk} \in \mathbb{R}^D$ and $\mathbf{R}_{sk} \in \mathbb{R}^{D \times D}$ denote the mean vector and precision matrix parameters of the Gaussian distribution, s and k denote utterance-level and frame-level mixture indexes, respectively, and S and K denote the number of speakers and the number of mixture components, respectively. Therefore, a set of model parameters Θ is defined as

$$\Theta \triangleq \{h_s, w_{sk}, \boldsymbol{\mu}_{sk}, \mathbf{R}_{sk} | s = 1, \cdots, S, k = 1, \cdots, K\}. \tag{8.70}$$

Note that this is almost equivalent to the following pdf of the GMM in Section 3.2.4:

$$p(\mathbf{O}, V|\Theta) = \prod_{t=1}^{T} w_{v_t} \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{v_t}, \mathbf{R}^{-1}_{v_t}). \tag{8.71}$$

The next section first describes Gibbs sampling for the standard GMM.

8.3.2 GMM marginal likelihood for complete data

We assume a diagonal covariance matrix for the Gaussian distributions as usual, where the d-d diagonal element of the precision/covariance matrix is expressed as r_d / Σ_d. The following conjugate distributions are used as the prior distributions of the model parameters:

$$p(\Theta|\Psi^0) : \begin{cases} \mathbf{w} \sim \mathrm{Dir}(\boldsymbol{\phi}^w), \\ \boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}^0_k, (\phi^\mu)^{-1} \mathbf{R}^{-1}_k), \\ r_{kd} \sim \mathrm{Gam}(\phi^r, r^0_{kd}), \end{cases} \tag{8.72}$$

where $\boldsymbol{\phi}^w, \boldsymbol{\mu}^0_k, \phi^\mu, r^0_{kd}, \phi^r (\triangleq \Psi^0)$ are the hyperparameters. $\mathrm{Dir}(\cdot)$ and $\mathrm{Gam}(\cdot)$ denote Dirichlet and gamma distributions in Appendices C.4 and C.11, respectively.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

In a Bayesian inference framework, we focus on the marginal likelihood for the complete data. In the complete data case, all of the latent variables are treated as observations, i.e., the assignments of all the latent variables are hypothesized to be given in advance. Then, $p(v_u = k|\cdot) \triangleq \delta(v_u, k)$ return 0 or 1 based on the assignment information, and the sufficient statistics of the GMM given all latent variables V can be represented as follows:

$$\begin{cases} \gamma_{V,k} &= \sum_t \delta(v_t, k), \\ \boldsymbol{\gamma}^{(1)}_{V,k} &= \sum_t \delta(v_t, k)\mathbf{o}_t, \\ \gamma^{(2)}_{V,kd} &= \sum_t \delta(v_t, k)(o_{td})^2. \end{cases} \tag{8.73}$$

The quantity $\gamma_{V,k} \in \mathbb{Z}^+$ is a count of frames assigned to k, and $\boldsymbol{\gamma}^{(1)}_{V,k}$ and $\gamma^{(2)}_{V,kd}$ are first-order and second-order sufficient statistics, respectively. We define a set of these sufficient statistics as

$$\Xi_{V,k} \triangleq \{\gamma_{V,k}, \boldsymbol{\gamma}^{(1)}_{V,k}, \gamma^{(2)}_{V,kd} | k = 1, \cdots, K, d = 1, \cdots, D\}. \tag{8.74}$$

The subscript V would be omitted when it is obvious. Note that the statistics γ_k is a positive discrete number while those appearing in the EM algorithm (ML in Eq. (3.153), MAP in Eq. (4.93), and VB in Eq. (7.67)) are positive continuous values since the occurrences of latent variables in the EM algorithm are expectation values based on the posterior probabilities of latent variables.

Based on the sufficient statistics representation in Eq. (8.73), the complete data likelihood of Eq. (8.71) can be represented by using the product formula of the Kronecker delta function in Eq. (A.4) ($f_b = \prod_a f_a^{\delta(a,b)}$) as follows:

$$\begin{aligned} p(\mathbf{O}, V|\Theta) &= \prod_{k=1}^{K}\prod_{t=1}^{T}\left(w_k^{\delta(v_t,k)}\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_k, \mathbf{R}_k^{-1})\right)^{\delta(v_t,k)} \\ &= \prod_{k=1}^{K}(w_k)^{\gamma_k}\prod_{t=1}^{T}\left(\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_k, \mathbf{R}_k^{-1})\right)^{\delta(v_t,k)}. \end{aligned} \tag{8.75}$$

Since the likelihood function is represented by the exponential distributions (multinomial and Gaussian distributions), these parameters integrate out, and we can use collapsed Gibbs sampling, as discussed in Section 8.1.4. The marginal likelihood for the complete data, $p(\mathbf{O}, V|\Psi^0)$, is represented by the following expectations by substituting Eqs. (8.72) and (8.75) into the following integration:

$$\begin{aligned} &p(\mathbf{O}, V|\Psi^0) \\ &= \int p(\mathbf{O}, V|\Theta)p(\Theta|\Psi^0)\mathrm{d}\Theta \\ &= \left(\mathbb{E}_{(\mathbf{w})}\left[\prod_{k=1}^{K}(w_k)^{\gamma_k}\right]\right)\left(\prod_{k=1}^{K}\mathbb{E}_{(\boldsymbol{\mu}_k,\mathbf{R}_k)}\left[\prod_{t=1}^{T}\left(\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_k, \mathbf{R}_k^{-1})\right)^{\delta(v_t,k)}\right]\right). \end{aligned} \tag{8.76}$$

Note that this calculation is similar to those of the MAP auxiliary function in Section 4.3.5 and variational lower bound in Section 7.3.4. For example, $\mathbb{E}_{(\mathbf{w})}\left[\prod_{k=1}^{K}(w_k)^{\gamma_k}\right]$ can be rewritten as

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{w})}\left[\prod_{k=1}^{K}(w_k)^{\gamma_k}\right] \\
&= \int p(\{w_k\}_{k=1}^K)\prod_{k=1}^{K}(w_k)^{\gamma_k}dw_k \\
&= \int \exp\left(\log\left(\mathrm{Dir}(\{w_k\}_{k=1}^K|\{\phi_k^w\}_{k=1}^K)\right)+\sum_{k=1}^{K}\gamma_k\log w_k\right)\prod_{k=1}^{K}dw_k. \qquad (8.77)
\end{aligned}$$

This is the same as the MAP auxiliary function of Eq. (4.54), and by analogy, Eq. (8.77) can be rewritten with the posterior distribution based equation in Eq. (4.54), as follows:

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{w})}\left[\prod_{k=1}^{K}(w_k)^{\gamma_k}\right] \\
&= \int \exp\left(\log\left(\mathrm{Dir}(\{w_k\}_{k=1}^K|\{\tilde{\phi}_k^w\}_{k=1}^K)\right)+\log\left(\frac{C_{\mathrm{Dir}}(\{\phi_k^w\}_{k=1}^K)}{C_{\mathrm{Dir}}(\{\tilde{\phi}_k^w\}_{k=1}^K)}\right)\right)\prod_{k=1}^{K}dw_k, \qquad (8.78)
\end{aligned}$$

where $\tilde{\phi}_s^w$ is a posterior hyperparameter, which is defined as:

$$\tilde{\phi}_k^w \triangleq \phi_k^w + \gamma_k. \qquad (8.79)$$

Therefore, the integral is finally solved as follows:

$$\begin{aligned}
\mathbb{E}_{(\mathbf{w})}\left[\prod_{k=1}^{K}(w_k)^{\gamma_k}\right] &= \frac{C_{\mathrm{Dir}}(\{\phi_k^w\}_{k=1}^K)}{C_{\mathrm{Dir}}(\{\tilde{\phi}_k^w\}_{k=1}^K)}\int \mathrm{Dir}(\{w_k\}_{k=1}^K|\{\tilde{\phi}_k^w\}_{k=1}^K)\prod_{k=1}^{K}dw_k \\
&= \frac{C_{\mathrm{Dir}}(\{\phi_k^w\}_{k=1}^K)}{C_{\mathrm{Dir}}(\{\tilde{\phi}_k^w\}_{k=1}^K)}. \qquad (8.80)
\end{aligned}$$

Thus, the integration is calculated analytically, similarly to the calculation of the variational lower bound. The integral with respect to $\boldsymbol{\mu}_k$ and $\mathbf{R}_k$ in Eq. (8.76) is also analytically calculated by using Eq. (7.124). First, the expectation is rewritten as follows:

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{\mu}_k,\mathbf{R}_k)}\left[\prod_{t=1}^{T}\left(\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_k,\mathbf{R}_k^{-1})\right)^{\delta(v_t,k)}\right] \\
&= \int \exp\Bigg(\sum_{d=1}^{D}\log\left(\mathcal{N}(\mu_{kd}|\tilde{\mu}_{kd},(\tilde{\phi}_k^{\mu}r_{kd})^{-1})\mathrm{Gam}_2(r_{kd}|\tilde{r}_{kd},\tilde{\phi}_k^r)\right) \\
&\quad + \left(-\frac{\gamma_k D}{2}\log(2\pi)+\frac{D}{2}\log\frac{\phi^{\mu}}{\tilde{\phi}_k^{\mu}}+\log\frac{C_{\mathrm{Gam}_2}(r_{kd}^0,\phi^r)}{C_{\mathrm{Gam}_2}(\tilde{r}_{kd},\tilde{\phi}_k^r)}\right)\Bigg)d\boldsymbol{\mu}_k d\mathbf{R}_k \\
&= \exp\left(-\frac{\gamma_k D}{2}\log(2\pi)+\frac{D}{2}\log\frac{\phi^{\mu}}{\tilde{\phi}_k^{\mu}}+\log\frac{C_{\mathrm{Gam}_2}(r_{kd}^0,\phi^r)}{C_{\mathrm{Gam}_2}(\tilde{r}_{kd},\tilde{\phi}_k^r)}\right), \qquad (8.81)
\end{aligned}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

where posterior hyperparameters $\tilde{\phi}_k^{\mu}$, $\tilde{\mu}_k$, $\tilde{\phi}_k^{r}$, and $\tilde{r}_{kd}$ are defined as follows:

$$\begin{cases} \tilde{\phi}_k^{\mu} & \triangleq \phi^{\mu} + \gamma_k, \\ \tilde{\mu}_k & \triangleq \frac{\phi^{\mu}\boldsymbol{\mu}_k^0 + \boldsymbol{\gamma}_k^{(1)}}{\tilde{\phi}_k^{\mu}}, \\ \tilde{\phi}_k^{r} & \triangleq \phi^{r} + \gamma_k, \\ \tilde{r}_{kd} & \triangleq r_{kd}^0 + \gamma_{kd}^{(2)} + \phi^{\mu}(\mu_{kd}^0)^2 - \tilde{\phi}_k^{\mu}(\tilde{\mu}_{kd})^2. \end{cases} \tag{8.82}$$

Thus, we finally solve all integrals in Eq. (8.76) with use of concrete forms of the normalization constants of the Dirichlet and gamma distributions in Appendices C.4 and C.11, as follows:

$$\begin{aligned} & p(\mathbf{O}, V|\Psi^0) \\ & = \frac{\Gamma(\sum_k \phi_k^w)}{\prod_k \Gamma(\phi_k^w)} \frac{\prod_k \Gamma(\tilde{\phi}_k^w)}{\Gamma(\sum_k \tilde{\phi}_k^w)} \prod_k (2\pi)^{-\frac{\gamma_k D}{2}} \frac{(\phi^{\mu})^{\frac{D}{2}} \left(\Gamma\left(\frac{\phi^r}{2}\right)\right)^{-D} \left(\prod_d \frac{r_{kd}^0}{2}\right)^{\frac{\phi^r}{2}}}{(\tilde{\phi}_k^{\mu})^{\frac{D}{2}} \left(\Gamma\left(\frac{\tilde{\phi}_k^r}{2}\right)\right)^{-D} \left(\prod_d \frac{\tilde{r}_{kd}}{2}\right)^{\frac{\tilde{\phi}_k^r}{2}}}. \end{aligned} \tag{8.83}$$

Below we summarize the posterior hyperparameters, which are obtained from the hyperparameters of the prior distributions (Ψ^0) and sufficient statistics (Eq. (8.73)) as follows:

$$\begin{cases} \tilde{\phi}_k^w & = \phi_k^w + \gamma_k, \\ \tilde{\phi}_k^{\mu} & = \phi^{\mu} + \gamma_k, \\ \tilde{\boldsymbol{\mu}}_k & = \frac{\phi^{\mu}\boldsymbol{\mu}_k^0 + \boldsymbol{\gamma}_k^{(1)}}{\tilde{\phi}_k^{\mu}}, \\ \tilde{\phi}_k^{r} & = \phi^{r} + \gamma_k, \\ \tilde{r}_{kd} & = r_{kd}^0 + \gamma_{kd}^{(2)} + \phi^{\mu}(\mu_{kd}^0)^2 - \tilde{\phi}_k^{\mu}(\tilde{\mu}_{kd})^2. \end{cases} \tag{8.84}$$

The marginal likelihood obtained is quite similar to the model parameter part of the variational lower bound in Eq. (7.126), since both functions are obtained by integrating out the Gaussian parameters for complete data likelihood. Based on the marginal likelihood for these complete data, we can calculate the marginal conditional distribution of v_t, as shown below.

8.3.3 GMM Gibbs sampler

As discussed in Section 8.1.4, a collapsed Gibbs sampler can assign latent variables by using the marginal conditional distribution. First, from the sum and product rules, the marginal conditional distribution $p(v_t = k|\mathbf{O}, V_{\setminus t})$ is represented as follows:

$$p(v_t = k|\mathbf{O}, V_{\setminus t}) = \frac{p(\mathbf{O}, V_{\setminus t}, v_t = k)}{p(\mathbf{O}, V_{\setminus t})} \propto p(\mathbf{O}, V_{\setminus t}, v_t = k|\Psi^0). \tag{8.85}$$

Here, $V_{\setminus t}$ indicates a set that does not include the tth frame element. Therefore, by considering the normalization constant, the posterior probability can be obtained by using Eq. (8.83) as follows:

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

$$p(v_t = k|\mathbf{O}, V_{\setminus t}) = \frac{p(\mathbf{O}, V_{\setminus t}, v_t = k|\Psi^0)}{\sum_{k'=1}^{K} p(\mathbf{O}, V_{\setminus t}, v_t = k'|\Psi^0)} = \frac{g(\tilde{\Psi}_{V_{\setminus t}, v_t=k})}{\sum_{k'=1}^{K} g(\tilde{\Psi}_{V_{\setminus t}, v_t=k'})}, \tag{8.86}$$

where $\tilde{\Psi}_{V_{\setminus t}, v_t=k}$ is a set of posterior hyperparameters computed given latent variables of $V_{\setminus t}, v_t = k$. Note that the denominators of $\tilde{\Psi}_{V_{\setminus t}, v_t=k'}$ have the same latent variable for all frames except t. To compute $\tilde{\Psi}_{V_{\setminus t}, v_t=k'}$, we first need to compute $\Xi_{V_{\setminus t},k}$, which is a set of sufficient statistics for all frames except t as follows:

$$\Xi_{V_{\setminus t},k} : \begin{cases} \gamma_{V_{\setminus t},k} & = \sum_{t'=\{1,\cdots,T\}\setminus t} \delta(v_{t'}, k), \\ \boldsymbol{\gamma}^{(1)}_{V_{\setminus t},k} & = \sum_{t'=\{1,\cdots,T\}\setminus t} \delta(v_{t'}, k)\mathbf{o}_t, \\ \gamma^{(2)}_{V_{\setminus t},kd} & = \sum_{t'=\{1,\cdots,T\}\setminus t} \delta(v_{t'}, k)(o_{td})^2. \end{cases} \tag{8.87}$$

From Eq. (8.73), $\Xi_{V_{\setminus t},k}$ can be computed by simply subtracting the zeroth-, first-, and second-order values of $v_t = k$ as:

$$\Xi_{V_{\setminus t},k} : \begin{cases} \gamma_{V_{\setminus t},k} & = \gamma_k - \delta(v_t, k), \\ \boldsymbol{\gamma}^{(1)}_{V_{\setminus t},k} & = \boldsymbol{\gamma}^{(1)}_k - \delta(v_t, k)\mathbf{o}_t, \\ \gamma^{(2)}_{V_{\setminus t},kd} & = \gamma^{(2)}_{kd} - \delta(v_{t'}, k)(o_{td})^2. \end{cases} \tag{8.88}$$

Thus, $\Xi_{V_{\setminus t},v_t=k'}$ can be obtained by simply adding the zeroth-, first-, and second-order values of $v_t = k'$ for all k' as:

$$\Xi_{V_{\setminus t},v_t=k'} : \begin{cases} \gamma_{V_{\setminus t},v_t=k'} & = \gamma_{V_{\setminus t},k'} + \delta(v_t, k'), \\ \boldsymbol{\gamma}^{(1)}_{V_{\setminus t},v_t=k'} & = \boldsymbol{\gamma}^{(1)}_{V_{\setminus t},k'} + \delta(v_t, k')\mathbf{o}_t, \\ \gamma^{(2)}_{V_{\setminus t},v_t=k',d} & = \gamma^{(2)}_{V_{\setminus t},k'd} + \delta(v_{t'}, k')(o_{td})^2. \end{cases} \quad \text{for all } k' \tag{8.89}$$

$g(\cdot)$ in Eq. (8.86) is defined as follows:

$$g(\tilde{\Psi}_k) \triangleq \Gamma(\tilde{\phi}^w_k)(\tilde{\phi}^\mu_k)^{-\frac{D}{2}} \left(\Gamma\left(\frac{\tilde{\phi}^r_k}{2}\right)\right)^D \left(\prod_d \frac{\tilde{r}_{kd}}{2}\right)^{-\frac{\tilde{\phi}^r_k}{2}}. \tag{8.90}$$

This equation is obtained by canceling out the factors in the numerator and denominator of Eq. (8.83). Thus, we obtain the Gibbs sampler, which assigns mixture component k at frame t.

Note that if we use multinomial distributions in LDA instead of Gaussian distributions, the numerator and denominator of the Gibbs sampler are further canceled out (see Griffiths & Steyvers (2004)) based on the formula of the gamma function in Appendix A.4. Actually, $\Gamma(\tilde{\phi}^w_k)$ in Eq. (8.90) can be similarly canceled out, while the other factors cannot be canceled out. Therefore, the computational cost of the Gaussian-based Gibbs sampler is large compared with LDA, since we need to compute Eq. (8.90) for every k and every frame t.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

8.3.4 Generative process and graphical model of multi-scale GMM

Based on solution of GMM Gibbs sampling in the previous section, we consider the Bayesian treatment of this multi-scale GMM for speaker clustering, which is an extension of GMM. For the conditional likelihood equation in Eq. (8.69), we again assume a diagonal covariance matrix for the Gaussian distributions. We also assume that the prior hyperparameters of the GMM parameters $\{w_{sk}\}_{sk}$, $\{\boldsymbol{\mu}_{sk}\}_{sk}$, and $\{\mathbf{R}_{sk}\}_{sk}$ for each s are shared with the parameters of one GMM (universal background model assumption (Reynolds *et al*. 2000)), which is used for speaker and speech recognition (subspace GMM (Povey, Burget, Agarwal *et al*. 2010)). Then the following conjugate distributions are used as the prior distributions of the model parameters:

$$p(\Theta|\Psi^0): \begin{cases} \mathbf{h} & \sim \mathrm{Dir}(\boldsymbol{\phi}^h), \\ \mathbf{w}_s & \sim \mathrm{Dir}(\boldsymbol{\phi}^w), \\ \boldsymbol{\mu}_{sk} & \sim \mathcal{N}(\boldsymbol{\mu}_k^0, (\phi^\mu)^{-1}\mathbf{R}_{sk}^{-1}), \\ r_{skd} & \sim \mathrm{Gam}(\phi^r, r_{kd}^0), \end{cases} \tag{8.91}$$

where $\boldsymbol{\phi}^h, \boldsymbol{\phi}^w, \boldsymbol{\mu}_k^0, \phi^\mu, r_{kd}^0, \phi^r (\triangleq \Psi^0)$ are the hyperparameters.

Based on the likelihood function and prior distributions, the generative process of multi-scale GMM can be expressed in Algorithm 17. The corresponding graphical model is shown in Figure 8.9. Now we have introduced multi-scale GMM, the following sections derive a solution for multi-scale GMM based on Gibbs sampling.

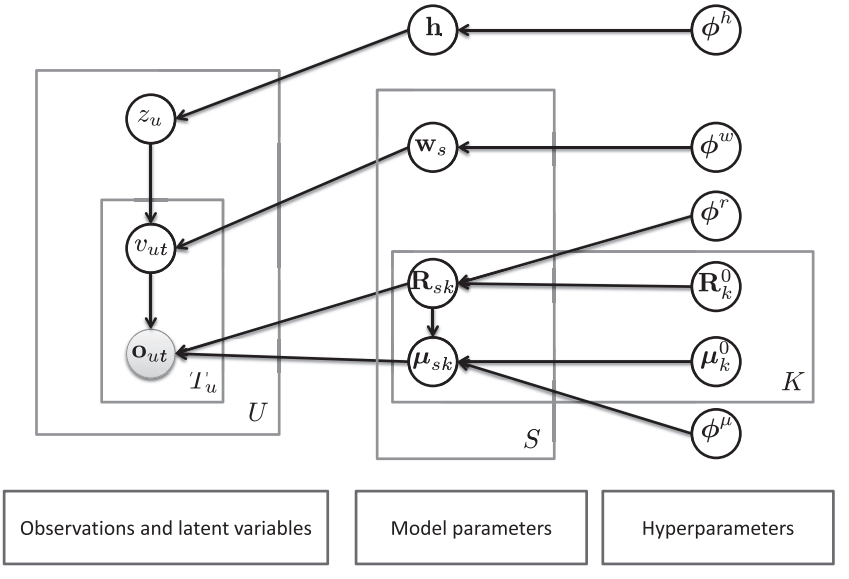


Figure 8.9 Model of multi-scale Gaussian mixture model. The model deals with two time scales based on frame t and utterance u.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

Algorithm 17 Generative process of multi-scale GMM

Require: Ψ^0

Draw $\mathbf{h}$ from Dir($\boldsymbol{\phi}^h$)
for each utterance-level mixture component $s = 1, \cdots, S$ **do**
  Draw $\mathbf{w}_s$ from Dir($\boldsymbol{\phi}^w$)
  for each frame-level mixture component $k = 1, \cdots, K$ **do**
    for each dimension $d = 1, \cdots, D$ **do**
      Draw r_{skd} from Gam(ϕ^r, r^0_{kd})
    end for
    Draw $\boldsymbol{\mu}_{sk}$ from $\mathcal{N}(\boldsymbol{\mu}^0_k, (\phi^\mu)^{-1}\mathbf{R}^{-1}_{sk})$
  end for
end for
for each utterance $u = 1, \cdots, U$ **do**
  Draw z_u from Mult ($\mathbf{h}$)
  for each frame $t = 1, \cdots, T_u$ **do**
    Draw v_{ut} from Mult ($\mathbf{w}_{z_u}$)
    Draw $\mathbf{o}_{ut}$ from $\mathcal{N}(\boldsymbol{\mu}_{z_u v_{ut}}, \mathbf{R}^{-1}_{z_u v_{ut}})$
  end for
end for

8.3.5 Marginal likelihood for the complete data

Similarly to the GMM Gibbs sampler, we prepare $p(v_{ut} = k|\cdot) \triangleq \delta(v_{ut}, k)$ given utterance u, which returns 0 or 1 based on the assignment information. In addition, we also prepare the assignment information of an utterance-level mixture with $p(z_u = s|\cdot) \triangleq \delta(z_u, s)$. The sufficient statistics of multi-scale GMM can be represented as follows:

$$\begin{cases} \xi_s & = \sum_u \delta(z_u, s), \\ \gamma_{sk} & = \sum_{u,t} \delta(z_u, s)\delta(v_{ut}, k), \\ \boldsymbol{\gamma}^{(1)}_{sk} & = \sum_{u,t} \delta(z_u, s)\delta(v_{ut}, k)\mathbf{o}_{ut}, \\ \gamma^{(2)}_{skd} & = \sum_{u,t} \delta(z_u, s)\delta(v_{ut}, k)(o_{utd})^2. \end{cases} \tag{8.92}$$

Here $\xi_s \in \mathbb{Z}^+$ is a count of utterances assigned to speaker cluster s, and $\gamma_{sk} \in \mathbb{Z}^+$ is a count of frames assigned to mixture component k in s. $\boldsymbol{\gamma}^{(1)}_{sk}$ and $\gamma^{(2)}_{skd}$ are first-order and second-order sufficient statistics, respectively.

Based on the sufficient statistics representation in Eq. (8.92), the complete data likelihood of Eq. (8.69) can be represented by using the product formula of the Kronecker delta function in Eq. (A.4) ($f_b = \prod_a f_a^{\delta(a,b)}$) as follows:

$$\begin{aligned} & p(\mathbf{O}, Z, V|\Theta) \\ & = \prod_{s=1}^{S}\prod_{u=1}^{U}\left(h_{z_u}\prod_{k=1}^{K}\prod_{t=1}^{T_u}\left(w_{z_u v_{ut}}\mathcal{N}(\mathbf{o}_{ut}|\boldsymbol{\mu}_{z_u v_{ut}}, \mathbf{R}^{-1}_{z_u v_{ut}})\right)^{\delta(v_{ut},k)}\right)^{\delta(z_u,s)} \\ & = \prod_{s=1}^{S}(h_s)^{\xi_s}\prod_{k=1}^{K}(w_{sk})^{\gamma_{sk}}\prod_{u=1}^{U}\prod_{t=1}^{T_u}\left(\mathcal{N}(\mathbf{o}_{ut}|\boldsymbol{\mu}_{sk}, \mathbf{R}^{-1}_{sk})\right)^{\delta(z_u,s)\delta(v_{ut},k)}. \end{aligned} \tag{8.93}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

The marginal likelihood for the complete data, $p(\mathbf{O}, Z, V|\Psi^0)$, is represented by the following expectations by substituting Eqs. (8.91) and (8.93) into the following integration:

$$\begin{aligned}
&p(\mathbf{O}, Z, V|\Psi^0) \\
&= \int p(\mathbf{O}, Z, V|\Theta) p(\Theta|\Psi^0) \mathrm{d}\Theta \\
&= \mathbb{E}_{(\mathbf{h})}\left[\prod_{s=1}^{S}(h_s)^{\xi_s}\right]\left(\prod_{s=1}^{S}\mathbb{E}_{(\mathbf{w}_s)}\left[\prod_{k=1}^{K}(w_{sk})^{\gamma_{sk}}\right]\right) \\
&\quad\times\left(\prod_{s=1}^{S}\prod_{k=1}^{K}\mathbb{E}_{(\boldsymbol{\mu}_{sk},\mathbf{R}_{sk})}\left[\prod_{u=1}^{U}\prod_{t=1}^{T_u}\left(\mathcal{N}(\mathbf{o}_{ut}|\boldsymbol{\mu}_{sk},\mathbf{R}_{sk}^{-1})\right)^{\delta(z_u,s)\delta(v_{ut},k)}\right]\right).
\end{aligned} \tag{8.94}$$

By following the derivations in Section 8.3.2, the expectations are calculated analytically. This section simply provides the analytical results of the expectations. By using Eq. (4.45), $\mathbb{E}_{(\mathbf{h})}\left[\prod_{s=1}^{S}(h_s)^{\xi_s}\right]$ is solved as:

$$\mathbb{E}_{(\mathbf{h})}\left[\prod_{s=1}^{S}(h_s)^{\xi_s}\right] = \frac{C_{\mathrm{Dir}}(\{\phi_s^h\}_{s=1}^S)}{C_{\mathrm{Dir}}(\{\tilde{\phi}_s^h\}_{s=1}^S)}, \tag{8.95}$$

where $\tilde{\phi}_s^h$ is a posterior hyperparameter, which is defined as:

$$\tilde{\phi}_s^h \triangleq \phi_s^h + \xi_s. \tag{8.96}$$

The other integral with respect to $\mathbf{w}_s$ in Eq. (8.94) is also calculated as follows:

$$\mathbb{E}_{(\mathbf{w}_s)}\left[\prod_{k=1}^{K}(w_{sk})^{\gamma_{sk}}\right] = \frac{C_{\mathrm{Dir}}(\{\phi_k^w\}_{k=1}^K)}{C_{\mathrm{Dir}}(\{\tilde{\phi}_{sk}^w\}_{k=1}^K)}, \tag{8.97}$$

where $\tilde{\phi}_{sk}^w$ is a posterior hyperparameter, which is defined as:

$$\tilde{\phi}_{sk}^w \triangleq \phi_k^w + \gamma_{sk}. \tag{8.98}$$

The integral with respect to $\boldsymbol{\mu}_{sk}$ and $\mathbf{R}_{sk}$ in Eq. (8.94) is also analytically calculated as follows:

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{\mu}_{sk},\mathbf{R}_{sk})}\left[\prod_{u=1}^{U}\prod_{t=1}^{T_u}\left(\mathcal{N}(\mathbf{o}_{ut}|\boldsymbol{\mu}_{sk},\mathbf{R}_{sk}^{-1})\right)^{\delta(z_u,s)\delta(v_{ut},k)}\right] \\
&= \exp\left(-\frac{\gamma_{sk}D}{2}\log(2\pi) + \frac{D}{2}\log\frac{\phi^\mu}{\tilde{\phi}_{sk}^\mu} + \log\frac{C_{\mathrm{Gam}_2}(r_{kd}^0, \phi^r)}{C_{\mathrm{Gam}_2}(\tilde{r}_{skd}, \tilde{\phi}_{sk}^r)}\right),
\end{aligned} \tag{8.99}$$

where posterior hyperparameters $\tilde{\phi}_{sk}^\mu$, $\tilde{\mu}_{sk}$, $\tilde{\phi}_{sk}^r$, and $\tilde{r}_{skd}$ are defined as follows:

$$\begin{cases}
\tilde{\phi}_{sk}^\mu &\triangleq \phi^\mu + \gamma_{sk}, \\
\tilde{\mu}_{sk} &\triangleq \frac{\phi^\mu \boldsymbol{\mu}_k^0 + \boldsymbol{\gamma}_{sk}^{(1)}}{\tilde{\phi}_{sk}^\mu}, \\
\tilde{\phi}_{sk}^r &\triangleq \phi^r + \gamma_{sk}, \\
\tilde{r}_{skd} &\triangleq r_{kd}^0 + \gamma_{skd}^{(2)} + \phi^\mu(\mu_{kd}^0)^2 - \tilde{\phi}_{sk}^\mu(\tilde{\mu}_{skd})^2.
\end{cases} \tag{8.100}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

Thus, we finally solve all the integrals in Eq. (8.94) as follows:

$$p(\mathbf{O}, Z, V|\Psi^0) = \frac{\Gamma(\sum_s \phi_s^h)}{\prod_s \Gamma(\phi_s^h)} \frac{\prod_s \Gamma(\tilde{\phi}_s^h)}{\Gamma(\sum_s \tilde{\phi}_s^h)} \prod_s \frac{\Gamma(\sum_k \phi_k^w)}{\prod_k \Gamma(\phi_k^w)} \frac{\prod_k \Gamma(\tilde{\phi}_{sk}^w)}{\Gamma(\sum_k \tilde{\phi}_{sk}^w)} \times \prod_{s,k} (2\pi)^{-\frac{\gamma_{sk} D}{2}} \frac{(\phi^\mu)^{\frac{D}{2}} \left(\Gamma\left(\frac{\phi^r}{2}\right)\right)^{-D} \left(\prod_d \frac{r_{kd}^0}{2}\right)^{\frac{\phi^r}{2}}}{(\tilde{\phi}_{sk}^\mu)^{\frac{D}{2}} \left(\Gamma\left(\frac{\tilde{\phi}_{sk}^r}{2}\right)\right)^{-D} \left(\prod_d \frac{\tilde{r}_{skd}}{2}\right)^{\frac{\tilde{\phi}_{sk}^r}{2}}}. \tag{8.101}$$

We summarize below the posterior hyperparameters, which are obtained from the hyperparameters of the prior distributions (Ψ^0) and sufficient statistics (Eq. (8.92)) as follows:

$$\begin{cases} \tilde{\phi}_s^h &= \phi_s^h + \xi_s, \\ \tilde{\phi}_{sk}^w &= \phi_k^w + \gamma_{sk}, \\ \tilde{\phi}_{sk}^\mu &= \phi^\mu + \gamma_{sk}, \\ \tilde{\boldsymbol{\mu}}_{sk} &= \frac{\phi^\mu \boldsymbol{\mu}_k^0 + \boldsymbol{\gamma}_{sk}^{(1)}}{\tilde{\phi}_{sk}^\mu}, \\ \tilde{\phi}_{sk}^r &= \phi^r + \gamma_{sk}, \\ \tilde{r}_{skd} &= r_{kd}^0 + \gamma_{skd}^{(2)} + \phi^\mu (\mu_{kd}^0)^2 - \tilde{\phi}_{sk}^\mu (\tilde{\mu}_{skd})^2. \end{cases} \tag{8.102}$$

Based on the marginal likelihood for these complete data, we can calculate the marginal conditional distribution of v_{ut} and z_u, as shown below.

8.3.6 Gibbs sampler

We provide a collapsed Gibbs sampler $p(v_{ut} = k|\mathbf{O}, V_{\setminus t}, Z_{\setminus u}, z_u = s)$ for a frame-level mixture component k, which has similarities with the GMM Gibbs sampler in Section 8.3.3. In addition, we also provide a collapsed Gibbs sampler $p(v_{u,t} = k'|\mathbf{O}, V_{\setminus t}, Z_{\setminus u}, z_u = s)$ for utterance-level mixture component s, which is a result of speaker clustering.

Frame-level mixture component

The Gibbs sampler assigns frame-level mixture component k at frame t by using the following equation:

$$\begin{aligned} p(v_{ut} = k|\mathbf{O}, V_{\setminus t}, Z_{\setminus u}, z_u = s) &\propto p(\mathbf{O}, V_{\setminus t}, v_{ut} = k, Z_{\setminus u}, z_u = s) \\ &\propto g(\tilde{\Psi}_{V_{\setminus t}, v_{ut}=k, Z_{\setminus u}, z_u=s}), \end{aligned} \tag{8.103}$$

where $g(\cdot)$ is defined as follows:

$$g(\tilde{\Psi}_{sk}) \triangleq \Gamma(\tilde{\phi}_{sk}^w) \left(\tilde{\phi}_{sk}^\mu\right)^{-\frac{D}{2}} \left(\Gamma\left(\frac{\tilde{\phi}_{sk}^r}{2}\right)\right)^D \left(\prod_d \frac{\tilde{r}_{sd}}{2}\right)^{-\frac{\tilde{\phi}_{sk}^r}{2}}. \tag{8.104}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

Algorithm 18 Gibbs sampling-based multi-scale mixture model.

```
Initialize Φ^0
repeat
  for u = shuffle (1 ··· U) do
    for t = shuffle (1 ··· T_u) do
      Sample v_{u,t} by using Eq. (8.105)
    end for
  end for
  for u = shuffle (1 ··· U) do
    Sample z_u by using Eq. (8.107)
  end for
until some condition is met
```

Therefore, by considering the normalization constant, the posterior probability can be obtained as follows:

$$p(v_{u,t} = k|\mathbf{O}, V_{\setminus t}, Z_{\setminus u}, z_u = s) = \frac{g(\tilde{\Psi}_{V_{\setminus t}, v_{ut}=k, Z_{\setminus u}, z_u=s})}{\sum_{k=1^K} g(\tilde{\Psi}_{V_{\setminus t}, v_{ut}=k', Z_{\setminus u}, z_u=s})}. \tag{8.105}$$

This equation is analytically derived by using the marginal likelihood for complete data (Eq. (8.101)).

Utterance-level mixture component

As with the frame-level mixture component case, the Gibbs sampler assigns utterance-level mixture s at utterance u by using the following equation:

$$\begin{aligned} p(z_u = s|\mathbf{O}, V, Z_{\setminus u}) &\propto p(\mathbf{O}, V, Z_{\setminus u}, z_u = s) \\ &\propto \frac{\Gamma(\sum_k \tilde{w}_{s\setminus u,k})}{\Gamma(\sum_k \tilde{w}_{s,k})} \prod_k g(\tilde{\Psi}_{sk}). \end{aligned} \tag{8.106}$$

The value of $\tilde{\Psi}_{s\setminus u,k}$ is computed by the sufficient statistics using $\mathbf{O}_{\setminus u}$ and $V_{\setminus u}$. Therefore, the posterior probability can be obtained as follows:

$$\begin{aligned} &p(z_u = s'|\mathbf{O}, V, Z_{\setminus u}) \\ &= \frac{\exp\left(\log \frac{\Gamma(\sum_k \tilde{w}_{s'\setminus u,k})}{\Gamma(\sum_k \tilde{w}_{s',k})} + \sum_k g_{s',k}(\tilde{\Psi}_{s',k}) - g_{s',k}(\tilde{\Psi}_{s'\setminus u,k})\right)}{\sum_{s,k} \exp\left(\log \frac{\Gamma(\sum_k \tilde{w}_{s\setminus u,k})}{\Gamma(\sum_k \tilde{w}_{s,k})} + g_{s,k}(\tilde{\Psi}_{s,k}) - g_{s,k}(\tilde{\Psi}_{s\setminus u,k})\right)}. \end{aligned} \tag{8.107}$$

Thus, we can derive a solution for the multi-scale mixture model based on Gibbs sampling, which jointly infers the latent variables by interleaving frame-level and utterance-level samples. Algorithm 18 provides a sample code for the proposed approach.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

Table 8.1 Comparison of MCMC and VB for speaker clustering. ACP: average cluster purity, ASP: average speaker purity, and K value: geometric mean of ACP and ASP.

Evaluation data	Method	ACP	ASP	K value
CSJ-1	MCMC	0.808	0.898	0.851
(# spkr10, # utt 50)	VB	0.704	0.860	0.777
CSJ-2	MCMC	0.852	0.892	0.871
(# spkr10, # utt 100)	VB	0.695	0.846	0.782
CSJ-3	MCMC	0.866	0.892	0.879
(# spkr10, # utt 200)	VB	0.780	0.870	0.823
CSJ-4	MCMC	0.784	0.694	0.738
(# spkr10, # utt 2,491)	VB	0.773	0.673	0.721
CSJ-5	MCMC	0.740	0.627	0.681
(# spkr10, # utt 2,321)	VB	0.693	0.676	0.684

MCMC-based acoustic modeling for speaker clustering was investigated with respect to the difference in the MCMC and VB estimation methods by Tawara, Ogawa, Watanabe *et al*. (2012). Table 8.1 shows speaker clustering results in terms of the average cluster purity (ACP), average speaker purity (ASP), and geometric mean of those values (K value) with respect to the evaluation criteria in speaker clustering. We used the Corpus of Spontaneous Japanese (CSJ) dataset (Furui *et al*. 2000) and investigated the speaker clustering performance for MCMC and VB for various amounts of data. Table 8.1 shows that the MCMC-based method outperformed the VB method by avoiding local optimum solutions, especially when only a few utterances could be used. These results also supported the importance of the Gibbs-based Bayesian properties.

Since the mixture of GMM is trained by MCMC, it is a straightforward extension to deal with a Dirichlet process mixture model, as discussed in Section 8.2.5, for speaker clustering, where the number of speaker clusters is jointly optimized based on this model. There are several studies of applying the Dirichlet process mixture model to speaker clustering (Fox *et al*. 2008, Tawara, Ogawa, Watanabe *et al*. 2012*b*). The next section introduces the application of an MCMC-based Dirichlet process mixture model to cluster HMMs (mixture of HMMs).

8.4 Nonparametric Bayesian HMMs to acoustic unit discovery

This section describes an application of Bayesian nonparametrics in Section 8.2 to acoustic unit discovery based on HMMs (Lee & Glass 2012, Lee, Zhang & Glass 2013, Torbati, Picone & Sobel 2013, Lee 2014). Acoustic unit discovery aims to automatically find an acoustic unit (e.g., phoneme) from speech data *without transcriptions*, and this is used to build ASR or spoken term detection systems with limited language resources (Schultz & Waibel 2001, Lamel, Gauvain & Adda 2002, Jansen, Dupoux, Goldwater *et al*. 2013). One of the powerful advantages of Bayesian nonparametrics is to find the model structure appropriately, and it is successfully applied to word unit

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

discovery in natural language processing (Goldwater, Griffiths & Johnson 2009, Mochihashi, Yamada & Ueda 2009) and in spoken language processing (Neubig, Mimura, Mori & Kawahara 2010). This section regards sub-word units as latent variables in one nonparametric Bayesian model. More specifically, it formulates a Dirichlet process mixture model where each mixture is an HMM used to model a sub-word unit and to generate observed segments of that unit. This model seeks the set of sub-word units, segmentation, clustering, and HMMs that best represents the observed data through an iterative inference process. This inference process can be performed by using Gibbs sampling.

To realize this acoustic unit discovery, the approach deals with the following variables:

- D dimensional speech feature $\mathbf{o}_t^n \in \mathbb{R}^D$ at frame t in utterance n.
- Binary boundary variable $b_t^n \in \{0, 1\}$ that has value 1 when the speech frame t is at the end point of a segment, and 0 otherwise.
- Boundary index $g_q^n = \{1, \cdots, t, \cdots, T^n\}$ returns the frame index of the qth boundary in utterance n. The initial boundary $g_0^n = 0$.
- Segment of features $\mathbf{O}_{t:t'}^n = \{\mathbf{o}_t^n, \cdots, \mathbf{o}_{t'}^n\}$.
- Unit label $c_{t:t'}^n \in \{1, \cdots, u, \cdots, U\}$ to specify the unit label of $\mathbf{O}_{t:t'}^n$. U is the number of the cluster, and u is a unit index. In addition, $c_t^n \in \{1, \cdots, u, \cdots, U\}$ indicates a unit label at frame t and utterance n.
- HMM Θ_u that represents one cluster unit u with a standard continuous density HMM (Section 3.2.3) that has state transition $a_{uij} \in [0, 1]$ from HMM state i to j, mixture weight $\omega_{ujk} \in [0, 1]$ at mixture component k in state j, and the mean vector $\boldsymbol{\mu}_{ujk} \in \mathbb{R}^D$ and (diagonal) precision matrix $\mathbf{R}_{ujk} \in \mathbb{R}^{D \times D}$.
- HMM state and GMM component $s_t^n \in \{1, \cdots, j, \cdots, J\}$ and $v_t^n \in \{1, \cdots, k, \cdots, K\}$.

The difference between this HMM and the conventional CDHMM in Section 3.2.3 given a phoneme unit is that this approach regards unit label $c_{t:t'}^n$ and the number of units U as a latent variable, which is obtained by a Dirichlet process. Therefore, the notation is similar to that in Section 3.2.3 except that it includes the cluster index u explicitly. A similar model is used in Gish, Siu, Chan *et al*. (2009) and Siu, Gish, Chan *et al*. (2014) based on an ML-style iterative procedure instead of Bayesian nonparametrics. In this section, the numbers of HMM states J and mixture components K are assumed to be the same fixed values for all units, but they can also be optimized by using Bayesian nonparametrics (Rasmussen 2000, Beal, Ghahramani & Rasmussen 2002, Griffiths & Ghahramani 2005).

8.4.1 Generative model and generative process

Since we use a fully Bayesian approach, the variables introduced in this model are regarded as probabilistic variables. For simplicity we consider that the boundary variable b_t^n is given in this formulation. The Bayesian approach first provides a generative process for complete data. We define latent variable Z as

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

$$Z \triangleq \{C, S, V, \{\gamma_u\}_{u=1}^{\infty}, \{\Theta_u\}_{u=1}^{\infty}\}. \tag{8.108}$$

Here, we assume the number of units is infinite, i.e., $U = \infty$, and latent variables will be generated based on the Dirichlet process. The other variables are defined as

$$\begin{aligned} C &\triangleq \{c_t^n | t = 1, \cdots, T_n, n = 1, \cdots, N\} \\ &= \{c_{(g_q+1):g_{q+1}}^n | q = 0, \cdots, Q_n, n = 1, \cdots, N\}, \\ S &\triangleq \{s_t^n | t = 1, \cdots, T_n, n = 1, \cdots, N\}, \\ V &\triangleq \{v_t^n | t = 1, \cdots, T_n, n = 1, \cdots, N\}. \end{aligned} \tag{8.109}$$

Q_n is the number of units appearing in utterance n, and $\zeta_u \in [0, 1]$ is a weight parameter of unit u. Then, the conditional distribution is represented as follows:

$$\begin{aligned} &p(\mathbf{O}, C, S, V | \{\zeta_u\}_{u=1}^{\infty}, \{\Theta_u\}_{u=1}^{\infty}) \\ &= \prod_{q=0}^{Q^n} p(c_{(g_q+1):g_{q+1}}^n = u | \{\zeta_u\}_{u=1}^{\infty}) p(\mathbf{o}_{g_q+1}^n, s_{g_q+1}^n, v_{g_q+1}^n | u, \Theta_u) \\ &\quad \times \prod_{t=g_q+2}^{g_{q+1}} p(\mathbf{o}_t^n, s_{t-1}^n, s_t^n, v_t^n | u, \Theta_u), \end{aligned} \tag{8.110}$$

where each likelihood function can be represented as follows:

$$p(c_{(g_q+1):g_{q+1}}^n = u | \{\zeta_u\}_{u=1}^{\infty}) = \zeta_u, \tag{8.111}$$

and

$$\begin{cases} p(\mathbf{o}_t^n, s_t^n = j, v_t^n = k | u, \Theta_u) = a_{uj} \omega_{ujk} \mathcal{N}(\mathbf{o}_t^n | \boldsymbol{\mu}_{ujk} \boldsymbol{\Sigma}_{ujk}) & (t = g_q + 1), \\ p(\mathbf{o}_t^n, s_{t-1}^n = i, s_t^n = j, v_t^n = k | u, \Theta_u) = a_{uij} \omega_{ujk} \mathcal{N}(\mathbf{o}_t^n | \boldsymbol{\mu}_{ujk} \boldsymbol{\Sigma}_{ujk}) & \text{Otherwise.} \end{cases} \tag{8.112}$$

a_{uj} is an initial weight in an HMM. For $p(\{\zeta_u\}_{u=1}^{\infty}, \{\Theta_u\}_{u=1}^{\infty})$, we use the Dirichlet process mixture model, described in Section 8.2.5.

The model parameters are sampled from a base distribution with hyperparameter Θ_0, and we use a conjugate prior distribution of CDHMM $p(\Theta_u | \Theta_0)$ with diagonal covariance matrices, as we discussed in Section 4.3.2, which is represented as

$$\begin{aligned} &p(\Theta_u) \\ &= p(\{a_{uj}\}_{j=1}^J) \left(\prod_{i=1}^J p(\{a_{uij}\}_{j=1}^J) \right) \left(\prod_{j=1}^J p(\{\omega_{ujk}\}_{k=1}^K) \right) \left(\prod_{j=1}^J \prod_{k=1}^K p(\boldsymbol{\mu}_{ujk}, \boldsymbol{\Sigma}_{ujk}) \right) \\ &= \mathrm{Dir}(\{a_{uj}\}_{j=1}^J | \phi^{\pi}) \left(\prod_{i=1}^J \mathrm{Dir}(\{a_{uij}\}_{j=1}^J | \phi^a) \right) \left(\prod_{j=1}^J \mathrm{Dir}(\{\omega_{ujk}\}_{k=1}^K | \phi^{\omega}) \right) \\ &\quad \times \left(\prod_{j=1}^J \prod_{k=1}^K \prod_{d=1}^D \mathcal{N}(\mu_{jkd} | \mu_d^0, (\phi^{\mu} r_{jkd})^{-1}) \mathrm{Gam}(r_{jkd} | r_d^0, \phi^r) \right). \end{aligned} \tag{8.113}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

ϕ^a, ϕ^ω, ϕ^μ, ϕ^r, $\boldsymbol{\mu}^0$, and $\mathbf{r}^0$ are the prior hyperparameter ($\triangleq \Theta_0$). $\boldsymbol{\mu}^0$ and $\mathbf{r}^0$ can be obtained using the Gaussian mean and precision parameters of all data.

Algorithm 19 provides a generative process of a nonparametric Bayesian HMM. A Dirichlet process mixture model (DPM) can sample the acoustic unit u from existing clusters or a new cluster for every speech segment. Thus, the model finally generates a sequence of speech features without fixing an acoustic unit explicitly.

Algorithm 19 Generative process of a nonparametric Bayesian HMM

Require: Concentration parameter γ, Base distribution of DP Θ_0, Boundary b_t^n.

for every utterance $n = 1, \cdots, N$ **do**
  for every segment $q = 1, \cdots, Q_n$ **do**
    Draw u and Θ_u from DPM($c^n_{(g_q+1):g_{q+1}}, \Theta_u|\gamma, \Theta_0$) (from existing clusters or a new one)
    Draw $\mathbf{a}_u$ from Dir($\boldsymbol{\phi}^\pi_u$)
    Draw $\mathbf{a}_{ui}$ from Dir($\boldsymbol{\phi}^a_{ui}$)
    Draw $\boldsymbol{\omega}_{uj}$ from Dir($\boldsymbol{\phi}^\omega_{uj}$)
    for every feature dimension $d = 1, \cdots, D$ **do**
      Draw r_{ujkd} from Gam(ϕ^r, r^0_d)
      Draw μ_{ujkd} from $\mathcal{N}(\mu^0_d, (\phi^\mu r_{ujkd})^{-1})$
    end for
    Draw j from Mult($s^n_{g_q+1}|\{a_{uj'}\}_{j'=1}^J$)
    Draw k from Mult($v^n_{g_q+1}|\{\omega_{ujk'}\}_{k'=1}^K$)
    Draw $\mathbf{o}$ from $\mathcal{N}(\mathbf{o}^n_{g_q+1}|\boldsymbol{\mu}_{ujk}, \boldsymbol{\Sigma}_{ujk})$
    for every frame $t = g_q + 2, \cdots, g_{q+1}$ **do**
      Draw j from Mult($s^n_t|\{a_{us_{t-1}j'}\}_{j'=1}^J$)
      Draw k from Mult($v^n_t|\{\omega_{ujk'}\}_{k'=1}^K$)
      Draw $\mathbf{o}$ from $\mathcal{N}(\mathbf{o}^n_t|\boldsymbol{\mu}_{ujk}, \boldsymbol{\Sigma}_{ujk})$
    end for
  end for
end for

8.4.2 Inference

The approach infers all latent variables by using Gibbs sampling, as discussed in Section 8.1.4, which samples a target latent variable z from the following conditional posterior distribution:

$$z \sim p(z|\mathcal{Z}_{\backslash z}, \mathbf{O}), \tag{8.114}$$

where $\mathcal{Z}_{\backslash z}$ denotes a set of all hidden variables in $\mathcal{Z}$ except for z. In this section we provide conditional distributions for cluster label $c^n_{t:t'}$, HMM state sequence $S^n_{t:t'}$, GMM component sequence $V^n_{t:t'}$, and HMM parameters Θ_u so that we can perform the Gibbs sampling of these latent variables.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

- Cluster label $c^n_{t:t'}$:
  Let $\mathcal{U}$ be the set of distinctive cluster units. The conditional posterior distribution of $c_{t:t'} = u \in \mathcal{U}$ is represented as follows:

$$\begin{aligned} p(c^n_{t:t'} = u|\cdots) &\propto p(c^n_{t:t'} = u|\mathcal{U}_{\backslash u}, \gamma) p(\mathbf{O}^n_{t:t'}|\Theta_u) \\ &= \frac{n^u}{N_u - 1 + \gamma} p(\mathbf{O}^n_{t:t'}|\Theta_u), \end{aligned} \tag{8.115}$$

  where γ is a hyperparameter of the DP prior, n^u represents the number of cluster labels in $\mathcal{U}_{\backslash u}$ taking the value u, and N_u is the number of speech segments. In this formulation, we do not marginalize Θ_u unlike Section 8.3, but use the sampled values, as discussed below.

  If $c^n_{t:t'}$ belongs to a new cluster that has not existed before, the conditional posterior distribution for this new cluster is represented as

$$p(c^n_{t:t'} \neq u, u \in \mathcal{U}|\cdots) \propto \frac{\gamma}{N_u - 1 + \gamma} \int p(\mathbf{O}^n_{t:t'}|\Theta) G(\Theta|\Theta_0) d\Theta, \tag{8.116}$$

  where the integral is approximated by a Monte Carlo estimation (Rasmussen 1999, Neal 2000, Tawara, Ogawa, Watanabe *et al*. 2012*b*). Note that Eqs. (8.115) and (8.116) are a typical solution of the Dirichlet process in Eq. (8.38). The Gibbs sampler for existing clusters $p(c^n_{t:t'} = u|\cdots)$ depends on the number of occurrences n^u and their likelihood, while that for a new cluster $p(c^n_{t:t'} \neq u, u \in \mathcal{U}|\cdots)$ depends on the concentration parameter γ and the marginalized likelihood.

  The likelihood values of $p(\mathbf{O}^n_{t:t'}|\Theta_u)$ and $p(\mathbf{O}^n_{t:t'}|\Theta)$ can be computed by considering all possible HMM states $S^n_{t:t'}$ and mixture components $V^n_{t:t'}$ based on the forward algorithm in Section 3.3.1. However, since the following Gibbs samplers can sample $S^n_{t:t'}$ and $V^n_{t:t'}$, we can use the following conditional likelihood values:

$$p(\mathbf{O}^n_{t:t'}|\Theta) \approx p(\mathbf{O}^n_{t:t'}|S^n_{t:t'}, V^n_{t:t'}, \Theta). \tag{8.117}$$

  This is easily computed by accumulating all Gaussian likelihood values given $S^n_{t:t'}$ and $V^n_{t:t'}$.
- HMM state s^n_t:
  The conditional posterior distribution of HMM state s^n_t is obtained from the following distribution, given the previous state s^n_{t-1} and the succeeding state s^n_{t+1}:

$$\begin{aligned} &p(s^n_t = j|\cdots) \\ &\quad \propto p(s^n_t = j|s^n_{t-1}) p(\mathbf{o}^n_t|\Theta_u, s^n_t = j) p(s^n_{t+1}|s^n_t = j) \\ &\quad = \begin{cases} a_{uj} \left(\sum_{k=1}^K \omega_{ujk} \mathcal{N}(\mathbf{o}^n_t|\boldsymbol{\mu}_{ujk}\boldsymbol{\Sigma}_{ujk}) \right) a_{ujs_{t+1}} & (t = g_q + 1) \\ a_{us_{t-1}j} \left(\sum_{k=1}^K \omega_{ujk} \mathcal{N}(\mathbf{o}^n_t|\boldsymbol{\mu}_{ujk}\boldsymbol{\Sigma}_{ujk}) \right) a_{ujs_{t+1}} & \text{otherwise.} \end{cases} \end{aligned} \tag{8.118}$$

  Note that this algorithm does not require the forward–backward algorithm in Section 3.3.1 to obtain the occupation probability, compared with the conventional HMM. There is an alternative algorithm to sample a state sequence similar to the forward-backward algorithm, called the forward filtering backward sampling algorithm of an HMM (Scott 2002, Mochihashi *et al*. 2009) in the MCMC framework.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

Given $v_t^n = k$, which is also obtained by the Gibbs sampler, we can also approximate Eq. (8.118) as

$$p(s_t^n = j|\cdots) \approx p(s_t^n = j|v_t^n = k, \cdots) \propto \begin{cases} a_{uj}\omega_{ujk}\mathcal{N}(\mathbf{o}_t^n|\boldsymbol{\mu}_{ujk}\boldsymbol{\Sigma}_{ujk})a_{u,j,s_{t+1}} & (t = g_q + 1) \\ a_{us_{t-1}j}\omega_{ujk}\mathcal{N}(\mathbf{o}_t^n|\boldsymbol{\mu}_{ujk}\boldsymbol{\Sigma}_{ujk})a_{u,j,s_{t+1}} & \text{otherwise.} \end{cases} \tag{8.119}$$

This avoids computing the Gaussian likelihoods for all mixture components.

- GMM component v_t^n:
  The conditional posterior distribution of GMM component $v_t^n = k$ at cluster u and state $s_t^n = j$ is obtained from the following distribution:

$$\begin{aligned} p(v_t^n = k|\cdots) &\propto p(v_t^n = k|\Theta_u, s_t^n = j)p(\mathbf{o}_t^n|\Theta_u, s_t^n = j, v_t^n = k) \\ &= \omega_{ujk}\mathcal{N}(\mathbf{o}_t^n|\boldsymbol{\mu}_{ujk}, \boldsymbol{\Sigma}_{ujk}). \end{aligned} \tag{8.120}$$

Thus, the Gibbs samplers of $p(c_{t:t'}^n|\cdots)$, $p(s_t^n|\cdots)$, and $p(v_t^n|\cdots)$ can provide the latent variable sequences of C, S, and V. Once we have C, S, and V, we can compute the sufficient statistics for the CDHMM, as follows:

$$\begin{cases} \xi_{ui} &= \sum_{n,q}\delta(c^n_{(g_q+1):g_{q+1}}, u)\delta(s^n_{g_q+1}, i), \\ \xi_{uij} &= \sum_{n,t}\delta(c_t^n, u)\delta(s_{t-1}^n, i)\delta(s_t^n, j), \\ \gamma_{ujk} &= \sum_{n,t}\delta(c_t^n, u)\delta(s_t^n, j)\delta(v_t^n, k), \\ \boldsymbol{\gamma}_{ujk}^{(1)} &= \sum_{n,t}\delta(c_t^n, u)\delta(s_t^n, j)\delta(v_t^n, k)\mathbf{o}_t^n, \\ \gamma_{ujkd}^{(2)} &= \sum_{n,t}\delta(c_t^n, u)\delta(s_t^n, j)\delta(v_t^n, k)(o_{td}^n)^2. \end{cases} \tag{8.121}$$

Thus, we can obtain the posterior distribution analytically based on the conjugate analysis, as we have shown in Sections 2.1.4 and 4.3. This section only provides the analytical solutions for the posterior distributions of CDHMM parameters, which are used to sample CDHMM parameters.

- HMM parameters Θ_u:
  - Initial weight

$$p(\mathbf{a}_u|\cdots) \propto \text{Dir}(\mathbf{a}_u|\boldsymbol{\phi}_u^{\pi}), \tag{8.122}$$

    where

$$\tilde{\phi}_{ui}^{\pi} = \phi^{\pi} + \xi_{ui}. \tag{8.123}$$

  - State transition

$$p(\mathbf{a}_{ui}|\cdots) \propto \text{Dir}(\mathbf{a}_{ui}|\boldsymbol{\phi}_{ui}^{a}), \tag{8.124}$$

    where

$$\tilde{\phi}_{uij}^{a} = \phi^{a} + \xi_{uij}. \tag{8.125}$$

  - Mixture weight

$$p(\boldsymbol{\omega}_{uj}|\cdots) \propto \text{Dir}(\boldsymbol{\omega}_{uj}|\boldsymbol{\phi}_{uj}^{\omega}), \tag{8.126}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

where

$$\tilde{\phi}^{\omega}_{ujk} = \phi^{\omega} + \gamma_{ujk}. \tag{8.127}$$

– Mean vector and covariance matrix at dimension d

$$\begin{aligned} &p(\mu_{ujkd}, r_{ujkd}| \cdots) \\ &\quad \propto \mathcal{N}(\mu_{ujkd}|\tilde{\mu}_{ujkd}, (\tilde{\phi}^{\mu}_{ujk} r_{ujkd})^{-1}) \mathrm{Gam}(r_{ujkd})|\tilde{\phi}^{r}_{ujk}, \tilde{r}_{ujkd}), \end{aligned} \tag{8.128}$$

where

$$\begin{cases} \tilde{\phi}^{\mu}_{ujk} & = \phi^{\mu} + \gamma_{ujk}, \\ \tilde{\boldsymbol{\mu}}_{ujk} & = \frac{\phi^{\mu}\boldsymbol{\mu}^0 + \boldsymbol{\gamma}^{(1)}_{ujk}}{\phi^{\mu} + \gamma_{ujk}}, \\ \tilde{\phi}^{r}_{ujk} & = \phi^{r} + \gamma_{ujk}, \\ \tilde{r}_{ujkd} & = \gamma^{(2)}_{ujkd} + \phi^{\mu}(\mu^0_d)^2 - \tilde{\phi}^{\mu}_{ujk}(\tilde{\mu}_{ujkd})^2 + r^0_d. \end{cases} \tag{8.129}$$

Note that compared with the collapsed Gibbs sampling solutions in Section 8.3, this samples the Gaussian parameters, as well as the other variables. Therefore, the Gibbs samplers for latent variables become rather simple equations.

Thus, we can obtain the nonparametric Bayesian HMMs for acoustic unit discovery given cluster boundaries. The MCMC is performed by sampling latent variables C, S, and V, and model parameters $\{\Theta_u\}_{u=1}^{U}$, iteratively. Note that the number of clusters U is changing according to the Dirichlet process, and finally becomes a fixed number. This clustering corresponds to automatically obtaining the acoustic unit in a Bayesian nonparametric manner.

Lee & Glass (2012) also considered cluster boundaries as latent variables which can be obtained by Gibbs sampling. In addition, to provide an appropriate initialization of the boundaries, the approach uses a pre-segmentation technique based on Glass (2003). The model obtained was compared with the other acoustic unit discovery method based on the dynamic time warping technique using the distance between GMM posteriors (Zhang & Glass 2009). The nonparametric Bayesian HMM achieved comparable results to the dynamic time warping technique in terms of the spoken term detection measures, which shows the effectiveness of the nonparametric Bayesian acoustic unit discovery. The approach carries a huge computational cost compared with the other EM type approaches (based on ML, MAP, and VB), and its scalability for the amount of data and the difficulty of parallelization remain serious problems. However, this is one of a few successful approaches using Bayesian nonparametrics for speech data, and it potentially has various advantages over the conventional EM type approaches (e.g., it could mitigate the local optimum problem and use any other distributions than conjugate distributions).

8.5 Hierarchical Pitman–Yor language model

In an LVCSR system, in addition to an acoustic model based on HMMs, the other key component is a language model based on n-grams. In recent years, BNP learning has

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

been substantially developed for language modeling (Teh *et al*. 2006) and successfully applied for several LVCSR tasks in Huang & Renals (2008). In this section, we introduce a hierarchical Bayesian interpretation for language modeling, based on a nonparametric prior called the Pitman–Yor (PY) process. The motivation of conducting Bayesian learning of n-gram language models is to tackle the limitations such as overfitting of maximum likelihood estimation and the lack of rich contextual knowledge sources. The PY process offers a principled approach to language model smoothing which produces a power-law distribution for natural language. The Bayesian language model based on Bayesian nonparametrics is a realization of a full Bayesian solution according to the MCMC inference procedure. Such a model is a *distribution estimate*, which is different from the Bayesian language model based on maximum a-posteriori (MAP) estimation as addressed in Section 4.7. A MAP-based language model is known as a *point estimate* of language model. A BNP-based language model integrates the values of parameters into a marginalized language model. It is interesting to note that the resulting hierarchical PY language model is a direct generalization of the *hierarchical Dirichlet language model*.

In what follows, we first survey the PY process and explain the importance of the power-law property in language modeling. Then we revisit language model smoothing based on Kneser–Ney smoothing and find the clue for connection to BNP learning. Next the hierarchical PY process is constructed to estimate a language model which provides the hierarchical Bayesian interpretation for a Kneser–Ney smoothed language model. The relation to the hierarchical Dirichlet language model will be illustrated. Lastly, the MCMC inference for the hierarchical PY language model is addressed.

8.5.1 Pitman–Yor process

The PY process (Pitman & Yor 1997) is known as the two-parameter Poisson–Dirichlet process $\mathrm{PY}(d, \alpha_0, G_0)$, which is expressed as a three-parameter distribution over distributions where $0 \leq d < 1$ is a discount parameter, α_0 is a strength parameter and G_0 is a base distribution. Base distribution G_0 can be understood as the mean of draws from the PY process. This process can be used to draw the unigram language model. Let $G(w)$ denote the unigram probability of a word $w \in \mathcal{V}$ and $G = [G(w)]_{w \in \mathcal{V}}$ denote the vector of unigram probabilities, which is drawn from a PY process:

$$G \sim \mathrm{PY}(d, \alpha_0, G_0). \tag{8.130}$$

Here, $G_0 = [G_0(w)]_{w \in \mathcal{V}}$ is a mean vector where $G_0(w)$ is the a-priori probability of word w. In practice, this base measure is usually set to be uniform $G_0 = 1/|\mathcal{V}|$ for all $w \in \mathcal{V}$. The parameters d and α_0 both control the degree of variability around G_0 in different ways. When $d = 0$, the PY process reverts to the DP, which is denoted by $\mathrm{DP}(\alpha_0, G_0)$. The PY process is seen as a generalization of the DP.

Basically, there is no analytic form for distribution of the PY process. We would like to work out the nonparametric distribution over sequences of words from the PY process. Let $\{w_1, w_2, \cdots\}$ be a sequence of words drawn independently and identically from G given the mean distribution G_0. The procedure of generating draws from a PY process

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

that can be described according to the metaphor of the "Chinese restaurant process" (Pitman 2006). As introduced in Section 8.2.4, imagine a Chinese restaurant containing an infinite number of tables, each with infinite seating capacity. Customers enter the restaurant and seat themselves. The first customer sits at the first table while each subsequent customer sits at an occupied table with probability proportional to the number of customers who are already sitting there $c_k - d$, or at a new unoccupied table with probability proportional to $\alpha_0 + dm.$ where $m.$ is the current number of occupied tables. That is, if z_i is the index of the table chosen by the ith customer, this customer sits at the table k given the seating arrangement of the previous $i-1$ customers, $\mathbf{z}^{-i} = \{z_1, \cdots, z_{i-1}\}$, with probability

$$p(z_i = k|\mathbf{z}^{-i}, d, \alpha_0) = \begin{cases} \frac{c_k - d}{\alpha_0 + c.} & 1 \le k \le m., \\ \frac{\alpha_0 + dm.}{\alpha_0 + c.} & k = m. + 1, \end{cases} \tag{8.131}$$

where c_k denotes the number of customers sitting at table k and $c. = \sum_k c_k$ is the total number of customers. The above generative procedure produces a sequence of words drawn independently from G with G marginalized out.

It is important to investigate the behaviors of drawing the sequence of words from the PY process. Firstly, the *rich-gets-richer* clustering property can be observed. That is, the more words have been assigned to a draw from G_0, the more likely subsequent words will be assigned to the draw. Secondly, the more we draw from G_0, the more likely a new word will be assigned to a new draw from G_0. Combining these two behaviors produces the so-called *power-law distribution* where many unique or distinct words are observed, most of them rarely. This distribution resembles the distribution of words which is seen in natural language. The power-law distribution has been found to be one of the most striking statistical properties of word frequencies in natural language. Figure 8.10 demonstrates the power-law behavior of the PY process which is controlled by parameters d and α_0, showing the average number of unique words among 25 sequences of words drawn from G, as a function of the number of words, for various values of α_0 and d. We find that α_0 controls the total number of unique words while d adjusts the asymptotic growth of the number of unique words. Figure 8.11 displays the proportion of words appearing only once among the unique words. These figures indicate the proportion of words that occur rarely. We can see that larger d and α_0 produce more rare words. This phenomenon is reflected by the probability of producing a new unoccupied table

$$p(z_i = k_{\text{new}}|\mathbf{z}^{-i}, d, \alpha_0) = \frac{\alpha_0 + dm.}{\alpha_0 + c.}. \tag{8.132}$$

8.5.2 Language model smoothing revisited

A key issue in a language model is to handle the sparseness of training data for training n-gram parameters under the conditions of a large n-gram window size n and large dictionary size $|\mathcal{V}|$. As addressed in Section 3.6, a series of language model smoothing methods has been developed to tackle the data sparseness issue in a statistical n-gram

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

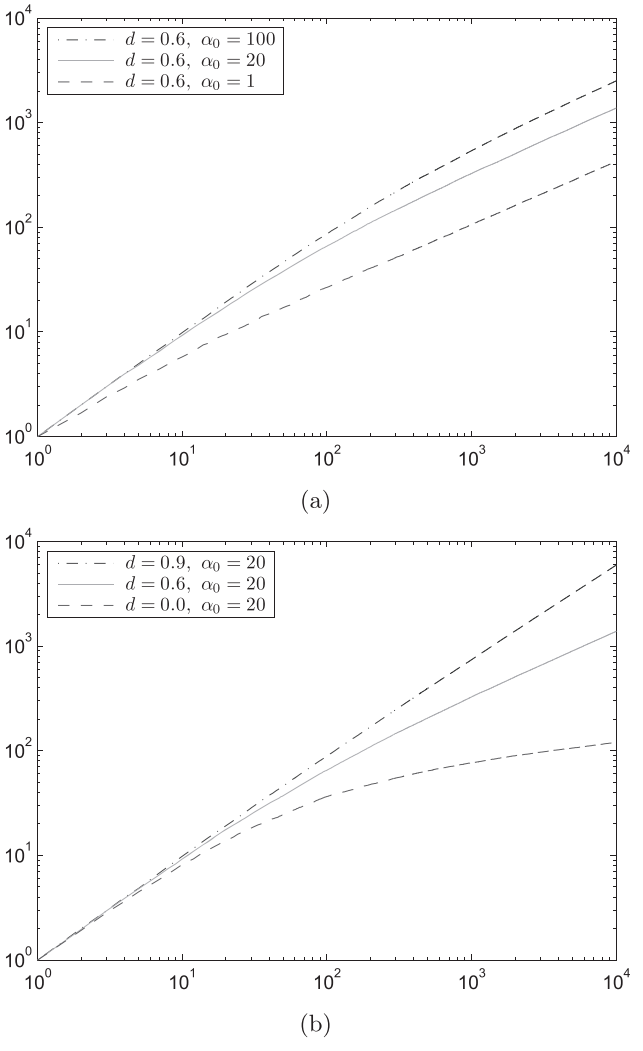
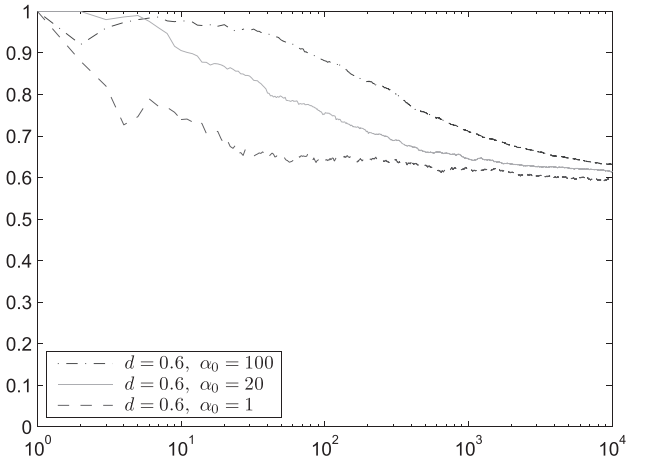


Figure 8.10 Power-law distribution: (a) the number of unique words as a function of the number of words, drawn on a log–log scale with $d = 0.6$ and $\alpha_0 = 100$ (dashdot line), 20 (solid line), and 1 (dashed line); (b) the same as (a) with $\alpha_0 = 20$ and $d = 0.9$ (dashdot line), 0.6 (solid line), and 0 (dashed line).

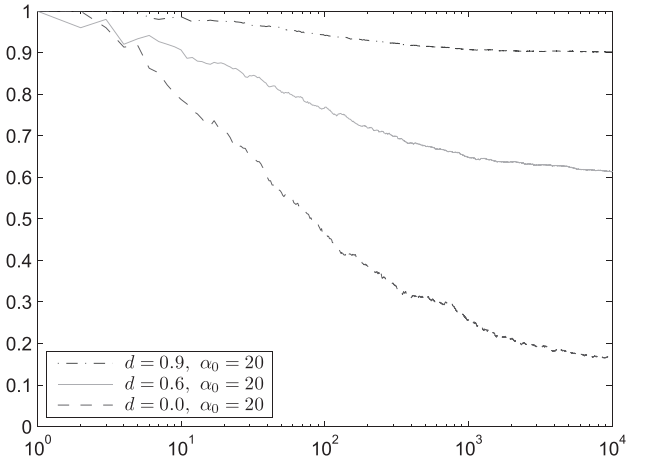
model. One important trick in these methods is to incorporate an absolute discount parameter d in the count of an observed n-gram event $w_{i-n+1}^{i} = \{w_i\, w_{i-n+1}^{i-1}\} \triangleq \{w\, \mathbf{u}\}$ of a word $w = w_i$ and its preceding history words $\mathbf{u} = w_{i-n+1}^{i-1}$. Owing to this discount, we modify the counts for lower order n-gram probabilities so as to construct the interpolated Kneser–Ney smoothing (Kneser & Ney 1995):

$$p_{\mathsf{KN}}(w|\mathbf{u}) = \frac{\max\{c_{\mathbf{u}w} - d_{|\mathbf{u}|}, 0\}}{c_{\mathbf{u}\cdot}} + \frac{d_{|\mathbf{u}|} m_{\mathbf{u}\cdot}}{c_{\mathbf{u}\cdot}} p_{\mathsf{KN}}(w|\pi(\mathbf{u})), \tag{8.133}$$

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press



(a)



(b)

Figure 8.11 Power-law distribution: (a) the proportion of words appearing only once, as a function of the number of words drawn with $d = 0.6$ and $\alpha_0 = 100$ (dashdot line), 20 (solid line), and 1 (dashed line); (b) the same as (a) with $\alpha_0 = 20$ and $d = 0.9$ (dashdot line), 0.6 (solid line), and 0 (dashed line).

which is rewritten from Eq. (3.232) with an order-dependent discount parameter $d \to d_{|\mathbf{u}|}$ and a new notation $N_{1+}(w_{i-n+1}^{i-1}, \bullet) \triangleq m_{\mathbf{u}\cdot} = |\{w' | c_{\mathbf{u}w'} > 0\}|$, expressing the number of unique words that follow the history words $\mathbf{u}$. The term $\pi(\mathbf{u})$ denotes the backoff context of $\mathbf{u}$. If $\mathbf{u} = w_{i-n+1}^{i-1}$, then $\pi(\mathbf{u}) = w_{i-n+2}^{i-1}$.

We would like to investigate the relation between language model smoothing and Bayesian learning. Such a relation is crucial to developing a BNP-based language model. In a standard Bayesian framework for language modeling, a prior distribution is placed over the predictive distribution for the language model. The predictive distribution is

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

estimated by marginalizing out the latent variables from the posterior distribution. The problem of zero-probability can be avoided by taking advantage of knowledge expressed by the priors. A smoothed language model can be obtained. In Yaman *et al*. (2007), the n-gram smoothing was conducted under a framework called the structural maximum a-posteriori (MAP) adaptation where the interpolation of n-gram statistics with an $n-1$-gram model was performed in a hierarchical and recursive fashion. As mentioned in Chapter 4, such a MAP approximation only finds a point estimate of a language model without considering the predictive distribution. To pursue a full Bayesian language model, the hierarchical Dirichlet language model (MacKay & Peto 1995) in Section 5.3 calculates the hierarchical predictive distribution of an n-gram by marginalizing the Dirichlet posterior over the Dirichlet priors.

However, the PY process was shown to be more fitted as a prior distribution than a Dirichlet distribution to the applications in natural language processing (Goldwater, Griffiths & Johnson 2006). It is because the *power-law* distributions of word frequencies produced by the PY process are more likely to be close to the *heavy-tailed distributions* observed in natural language. But, the PY process, addressed in Section 8.5.1, is only designed for a unigram language model. In what follows, we extend the hierarchical Dirichlet language model, which adopts the Dirichlet prior densities, to the hierarchical PY language model, which utilizes the PY process as nonparametric priors and integrates out these prior measures.

8.5.3 Hierarchical Pitman–Yor language model

Similarly to the extension from the Dirichlet process to the hierarchical Dirichlet process for representation of multiple documents, in this section, we address the extension from the PY process to the hierarchical PY (HPY) process which is developed to realize the probability measure for the smoothed n-gram language model. Let $G_\emptyset = [G_\emptyset(w)]_{w\in\mathcal{V}}$ represent the vector of word probability estimates for unigrams of all words w in a vocabulary $\mathcal{V}$. A PY process prior for unigram probabilities is expressed by

$$G_\emptyset \sim \mathrm{PY}(d_0, \alpha_0, G_0), \tag{8.134}$$

where G_0 is a global mean vector in the form of a noninformative base distribution or uniform distribution:

$$G_0(w) = \frac{1}{|\mathcal{V}|} \quad \text{for all } w \in \mathcal{V}. \tag{8.135}$$

This PY process can be used to calculate the predictive unigram of a word w according to the metaphor of the Chinese restaurant process. The customers or word tokens enter the restaurant and seat themselves at either an occupied table or a new table with the probabilities in Eq. (8.131). Each table is labeled by a word $w \in \mathcal{V}$ initialized by the first customer sitting on it. The next customer can only sit on those tables with the same label w. That is, those c_w customers corresponding to the same word label w can sit at different tables, with m_w being the number of tables with label w. In our notation, $m_\cdot = \sum_k m_k$ means the total number of tables. The number of customers at table k is denoted by c_k, and the total number of customers is expressed by $c_\cdot = \sum_k c_k$. Given

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

the seating arrangement of customers S, the discount parameter d_0, and the strength parameter α_0, the predictive unigram probability of a new word w is given by

$$p(w|S, d_0, \alpha_0) = \sum_{k=1}^{m.} \frac{c_k - d_0}{\alpha_0 + c.} \delta(k, w) + \frac{\alpha_0 + d_0 m.}{\alpha_0 + c.} G_0(w)$$
$$= \frac{c_w - d_0 m_w}{\alpha_0 + c.} + \frac{\alpha_0 + d_0 m.}{\alpha_0 + c.} G_0(w), \tag{8.136}$$

where $\delta(k, w) = 1$ if table k has the label w, $\delta(k, w) = 0$ otherwise. This equation is derived similarly to the metaphor of the Chinese restaurant process designed for the Dirichlet process, as mentioned in Section 8.2.4. The key difference of the PY process compared to the Dirichlet process is the discount parameter d_0 which is used to adjust the power-law distribution of unigrams based on the PY process. By averaging over seating arrangements and hyperparameters (S, d_0, α_0), we obtain the probability $p(w)$ for a unigram language model.

Interestingly, if we set $d_0 = 0$, the PY process is reduced to a DP which produces the Dirichlet distribution $\mathrm{Dir}(\alpha_0 G_0)$. In this case, the predictive unigram probability based on the PY process prior in Eq. (8.136) is accordingly reduced to

$$p(w|S, \alpha_0) = \frac{c_w}{\alpha_0 + c.} + \frac{\alpha_0 G_0(w)}{\alpha_0 + c.}$$
$$= \frac{c_w + \frac{\alpha_0}{|\mathcal{V}|}}{\sum_{w \in \mathcal{V}} [c_w + \frac{\alpha_0}{|\mathcal{V}|}]}, \tag{8.137}$$

where $G_0(w) = 1/|\mathcal{V}|$ is used for all w. This equation is equivalent to the hierarchical Dirichlet unigram model as given in Eq. (5.70). The only difference here is to adopt a single shared hyperparameter α_0 for all word labels $w \in \mathcal{V}$.

Next, we extend the unigram language model to the n-gram language model based on the HPY process. An n-gram language model is defined as a probability measure over the current word $w = w_i$ given a history context $\mathbf{u} = w_{i-n+1}^{i-1}$. Let $G_{\mathbf{u}} = [G_{\mathbf{u}}(w)]_{w \in \mathcal{V}}$ be the vector of the target probability distributions of all vocabulary words $w \in \mathcal{V}$ given the history context $\mathbf{u}$. We use a PY process as the prior for $G_{\mathbf{u}}[G_{\mathbf{u}}(w)]_{w \in \mathcal{V}}$ in the form of

$$G_{\mathbf{u}} \sim \mathrm{PY}(d_{|\mathbf{u}|}, \alpha_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}), \tag{8.138}$$

with the hyperparameters $d_{|\mathbf{u}|}$ and $\alpha_{|\mathbf{u}|}$ specific to the length of context $|\pi(\mathbf{u})|$. However, $G_{\pi(\mathbf{u})}$ is still an unknown base measure. A PY process is recursively placed over it:

$$G_{\pi(\mathbf{u})} \sim \mathrm{PY}(d_{|\pi(\mathbf{u})|}, \alpha_{|\pi(\mathbf{u})|}, G_{\pi(\pi(\mathbf{u}))}), \tag{8.139}$$

with parameters which are functions of $|\pi(\pi(\mathbf{u}))|$. This is repeated until we reach the PY process prior $G_\emptyset$ for a unigram model. Figure 8.12 is a schematic diagram showing the hierarchical priors for the smoothed language model based on the HPY process. This process enables us to generalize from the unigram language model to the n-gram language model. The resulting probability distribution is called the hierarchical Pitman–Yor language model (HPYLM).

A hierarchical Chinese restaurant process can be used to develop a generative procedure for drawing words from the HPYLM with all $G_{\mathbf{u}}$ marginalized out. The context

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

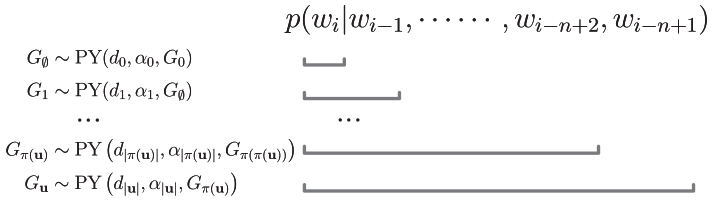


Figure 8.12 Hierarchical Pitman–Yor process for n-gram language model.

$\mathbf{u}$ corresponds to a restaurant. This procedure gives us a representation of HPYLM for efficient inference using an MCMC algorithm, and easy computation of the predictive probability distribution from new test words. Through this representation, the correspondence between the Kneser–Ney language model and HPYLM can be illustrated. In the metaphor of hierarchical CRP, there are multiple hierarchical restaurants or PY processes, with each corresponding to one context. Different orders of n-gram models share information with each other through the interpolation of higher-order n-grams with lower-order n-grams. We draw words from $G_{\mathbf{u}}$ of the PY process by using the CRP as discussed in Section 8.5.1. Further, we draw words from another CRP to sample the parent distribution $G_{\pi(\mathbf{u})}$, which is itself sampled according to a PY process. This is recursively applied until we need draws from the global mean distributions G_0. By referring to the predictive unigram probability from the PY process as shown in Eq. (8.136), the predictive n-gram probability $p(w|\mathbf{u}, S, d_{|\mathbf{u}|}, \alpha_{|\mathbf{u}|})$ under a particular combination of seating arrangement S and the hyperparameters $d_{|\mathbf{u}|}$ and $\alpha_{|\mathbf{u}|}$ can be obtained from

$$p(w|\mathbf{u}, S, d_{|\mathbf{u}|}, \alpha_{|\mathbf{u}|}) = \frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|} m_{\mathbf{u}w}}{\alpha_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \frac{\alpha_{|\mathbf{u}|} + d_{|\mathbf{u}|} m_{\mathbf{u}\cdot}}{\alpha_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} p(w|\pi(\mathbf{u}), S, d_{|\pi(\mathbf{u})|}, \alpha_{|\pi(\mathbf{u})|}), \tag{8.140}$$

where $c_{\mathbf{u}wk}$ is the number of customers sitting at table k with label w, $c_{\mathbf{u}w\cdot} = \sum_k c_{\mathbf{u}wk}$ and $m_{\mathbf{u}w}$ is the number of occupied tables with label w. It is interesting to see that the Kneser–Ney language model (KN–LM) in Eq. (8.133) is closely related to the HPYLM in Eq. (8.140). HPYLM is a generalized realization of KN–LM with an additional concentration parameter $\alpha_{|\mathbf{u}|}$. We can interpret the interpolated KN–LM as an approximate inference scheme for the HPYLM.

8.5.4 MCMC inference for HPYLM

An MCMC algorithm can be used to infer the posterior probability of seating arrangement S. Given some training data $\mathcal{D}$, we count the number of occurrences $c_{\mathbf{u}w\cdot}$ of each word w appearing after each context $\mathbf{u}$ of length $n-1$. This means that there are $c_{\mathbf{u}w\cdot}$ samples drawn from the PY process $G_{\mathbf{u}}$. We are interested in the posterior probability over the latent variables $\mathcal{G} = \{G_{\mathbf{u}} \mid \text{all contexts } \mathbf{u}\}$ and the parameters $\Theta = \{d_m, \alpha_m\}$ of all lower-order models with $0 \leq m \leq n-1$. Because the hierarchical Chinese restaurant process marginalizes out each $G_{\mathbf{u}}$, we can replace $\mathcal{G}$ by the seating arrangement in the

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

corresponding restaurant using $S = \{S_{\mathbf{u}} \mid \text{all contexts } \mathbf{u}\}$. The posterior probability is then obtained from

$$p(S, \Theta|\mathcal{D}) = \frac{p(S, \Theta, \mathcal{D})}{p(\mathcal{D})}. \tag{8.141}$$

Given this posterior probability, we can calculate the predictive n-gram probability of a test word w after observing a context $\mathbf{u}$:

$$\begin{aligned} p(w|\mathbf{u}, \mathcal{D}) &= \int p(w|\mathbf{u}, S, \Theta) p(S, \Theta|\mathcal{D}) d(S, \Theta) \\ &= \mathbb{E}_{(S,\Theta)}[p(w|\mathbf{u}, S, \Theta)] \\ &\approx \frac{1}{L} \sum_{l=1}^{L} p(w|\mathbf{u}, S^{(l)}, \Theta^{(l)}), \end{aligned} \tag{8.142}$$

which is an expectation of predictive probability under a particular set of seating arrangements S and PY parameters Θ. The overall predictive probability in the integral of Eq. (8.142) is approximated by using the L samples $\{S^{(l)}, \Theta^{(l)}\}$ drawn from posterior probability $p(S, \Theta|\mathcal{D})$.

In the implementation of Teh *et al*. (2006) and Huang & Renals (2008), the discount parameter and concentration parameter were drawn by a beta distribution and a gamma distribution, respectively. Here, we address the approach to sampling the seating arrangement $S_{\mathbf{u}}$ corresponding to $G_{\mathbf{u}}$. We employ Gibbs sampling to keep track of the current state of each variable of interest in the model, and iteratively re-sample the state of each variable given the current states of all other variables. After a sufficient number of iterations, the states of variables in the seating arrangement S converge to the required samples from the posterior probability. The variables consist of, for each context (restaurant) $\mathbf{u}$ and each word (customer) $x_{\mathbf{u}l}$ drawn from $G_{\mathbf{u}}$, the index $k_{\mathbf{u}l}$ of the draw from $G_{\pi(\mathbf{u})}$ assigned $x_{\mathbf{u}l}$. In the Chinese restaurant metaphor, this is the table index of where the lth customer sat at the restaurant corresponding to $G_{\mathbf{u}}$. If $x_{\mathbf{u}l}$ has value w, it can only be assigned to draws from $G_{\pi(\mathbf{u})}$ that have the same value w. The posterior probability of drawing a table $k_{\mathbf{u}l}$ for the last word $x_{\mathbf{u}l}$ from $G_{\mathbf{u}}$ is given below:

$$p(k_{\mathbf{u}l} = k|S^{-\mathbf{u}l}, \Theta) \propto \frac{\max(0, c_{\mathbf{u}x_{\mathbf{u}l}k}^{-\mathbf{u}l} - d)}{\alpha + c_{\mathbf{u}\cdot\cdot}^{-\mathbf{u}l}}, \tag{8.143}$$

$$p(k_{\mathbf{u}l} = k_{\text{new}}|S^{-\mathbf{u}l}, \Theta) \propto \frac{\alpha + d m_{\mathbf{u}\cdot}^{-\mathbf{u}l}}{\alpha + c_{\mathbf{u}\cdot\cdot}^{-\mathbf{u}l}} p(x_{ul}|\pi(\mathbf{u}), S^{-\mathbf{u}l}, \Theta), \tag{8.144}$$

where the superscript $-\mathbf{u}l$ means the corresponding set of variables or counts with $x_{\mathbf{u}l}$ excluded.

One other key difference between the KN–LM in Eq. (8.133) and the HPYLM in Eq. (8.140) is that the KN–LM adopts a fixed discount $d_{|\mathbf{u}|}$ while HPYLM produces different discounts $d_{|\mathbf{u}|} m_{\mathbf{u}w}$ for different words w due to the values of $m_{\mathbf{u}w}$, which counts the number of occupied tables labeled by w. In general, $m_{\mathbf{u}w}$ is on average larger if $c_{\mathbf{u}w}$ is larger. The actual amount of discount grows gradually as the count $c_{\mathbf{u}w}$ grows. The physical meaning of discounting in the smoothed n-grams based on HPYLM is consistent

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press

with the discount scheme in the modified Kneser–Ney language model (MKN–LM), which specifies three discounts d_1, d_2, and d_{3+} for those n-grams with one ($c = 1$), two ($c = 2$), and three or more ($c \geq 3$) counts, as addressed in Section 3.6.6. The discounts $\{d_1, d_2, d_{3+}\}$ in MKN–LM are empirically determined while the discounts $d_{|\mathbf{u}|}m_{\mathbf{u}w}$ in HPYLM are automatically associated with each word w. In particular, if we restrict $m_{\mathbf{u}w\cdot}$ to be at most 1,

$$m_{\mathbf{u}w} = \min(1, c_{\mathbf{u}w\cdot}), \tag{8.145}$$

we get the same discount value so long as $c_{\mathbf{u}w\cdot} > 0$, or equivalently we conduct absolute discounting. We also have the relationships among the $c_{\mathbf{u}w}$s and $m_{\mathbf{u}w}$:

$$c_{\mathbf{u}w} = \sum_{\mathbf{u}':\pi(\mathbf{u}')=\mathbf{u}} m_{\mathbf{u}'w\cdot}. \tag{8.146}$$

If we further assume $\alpha_{|\mathbf{u}|} = \alpha_m = 0$ for all model orders $0 \leq m \leq n-1$, the predictive probability of HPYLM in Eq. (8.140) is directly reduced to that given by the interpolated Kneser–Ney model.

8.6 Summary

Due to the high computational cost and lack of scalability for data and model sizes, MCMC approaches are still in a development stage for speech and language processing applications that essentially need to deal with large-scale data. However, the recent advance in computational powers (CPU speed, memory size, many cores, and GPU) and algorithm development make it possible to realize middle-scale data processing of MCMC, especially for language processing based on multinomial and Dirichlet distributions, as discussed in this chapter. One of the most attractive features of MCMC is that we can handle *any types of expectation calculation* by the Monte Carlo approximations, which can realize all Bayesian approaches in principle. Therefore, we expect further development of computational powers and algorithms to widen the applications of MCMC to large-scale problems, and enable fully Bayesian speech and language processing based on MCMC in the near future.

https://doi.org/10.1017/CBO9781107295360.009 Published online by Cambridge University Press