6 Asymptotic approximation

Asymptotic approximation is also well known in practical Bayesian approaches (De Bruijn 1970) for approximately obtaining the posterior distributions. For example, as we discussed in Chapter 2, the posterior distributions of a model parameter $p(\Theta|\mathbf{O})$ and a model $p(M|\mathbf{O})$ given an observation $\mathbf{O} = \{\mathbf{o}_t \in \mathbb{R}^D | t = 1, \cdots, T\}$) are usually difficult to solve. The approach assumes that we have enough data (i.e., T is sufficiently large), which also makes Bayesian inference mathematically tractable. As a particular example of asymptotic approximations, we introduce the Laplace approximation and Bayesian information criterion, which are widely used for speech and language processing.

The Laplace approximation is used to approximate a complex distribution as a Gaussian distribution (Kass & Raftery 1995, Bernardo & Smith 2009). It assumes that the posterior distribution is highly peaked at about its maximum value, which corresponds to the mode of the posterior distribution. Then the posterior distribution is modeled as a Gaussian distribution with the mode as a mean parameter. By using the approximation, we can obtain the posterior distributions analytically to some extent. Section 6.1 first explains the Laplace approximation in general. In Sections 6.3 and 6.4 we also discuss use of the Laplace approximation for analytically obtaining Bayesian predictive distributions for acoustic modeling and Bayesian extension of successful neural-network-based acoustic modeling, respectively.

Another example of this asymptotic approximation is the Bayesian information criterion (or Schwarz criterion (Schwarz 1978)). The Bayesian information criterion also assumes the large sample case, and approximates the posterior distribution of a model $p(M|\mathbf{O})$ with a simple equation. Since the Bayesian information criterion assumes the large sample case, it is also described as an instance of asymptotic approximations. Section 6.2 explains the Bayesian information criterion in general; it is used for model selection problems in speech processing. For example, Section 6.5 discusses the optimization of an appropriate model structure of hidden Markov models, and Section 6.6 discusses estimation of the number of speakers and detecting speech segments in conversations by regarding these approaches as model selection problems.

6.1 Laplace approximation

This section first describes the basic theory of the Laplace approximation. We first consider a simple case where a model does not have a latent variable. We focus on

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

the posterior distributions of model parameters, but this approximation can be applied to the other continuous probabilistic variables. Let $\boldsymbol{\theta} \in \mathbb{R}^J$ be a J dimensional vector form of continuous model parameters and $\mathbf{O} = \{\mathbf{o}_t \in \mathbb{R}^D | t = 1, \cdots, T\}$ be a T-frame sequence of D dimensional feature vectors. We consider the posterior distribution of $\boldsymbol{\theta}$, which has the following distribution $p(\boldsymbol{\theta}|\mathbf{O})$:

$$p(\boldsymbol{\theta}|\mathbf{O}) \triangleq \exp(-f(\boldsymbol{\theta})), \tag{6.1}$$

where $f(\boldsymbol{\theta})$ is a continuous function over $\boldsymbol{\theta}$. Note that most parametric distributions can be represented by this equation. The Laplace approximation approximates $p(\boldsymbol{\theta}|\mathbf{O})$ as a Gaussian distribution with the mode of $p(\boldsymbol{\theta}|\mathbf{O})$ as a mean parameter, which can be obtained numerically or analytically for some specific distributions. Let $\boldsymbol{\theta}^{\text{MAP}}$ be the mode (MAP value, as discussed in Chapter 4) of $p(\boldsymbol{\theta}|\mathbf{O})$, that is:

$$\begin{aligned} \boldsymbol{\theta}^{\text{MAP}} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{O}) \\ &= \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}). \end{aligned} \tag{6.2}$$

Here we use Eq. (6.1). Since the mode $\boldsymbol{\theta}^{\text{MAP}}$ is the minimum value in $f(\boldsymbol{\theta})$, $\boldsymbol{\theta}^{\text{MAP}}$ also satisfies the following equation:

$$\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{MAP}}} = 0, \tag{6.3}$$

where $\nabla_{\boldsymbol{\theta}}$ is the gradient operator with respect to $\boldsymbol{\theta}$. Based on the property, a Taylor expansion around $\boldsymbol{\theta}^{\text{MAP}}$ approximates $f(\boldsymbol{\theta})$ as follows:

$$\begin{aligned} f(\boldsymbol{\theta}) &\approx f(\boldsymbol{\theta}^{\text{MAP}}) + (\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{MAP}})^{\mathsf{T}} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{MAP}}} + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{MAP}})^{\mathsf{T}}\mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{MAP}}) \\ &= f(\boldsymbol{\theta}^{\text{MAP}}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{MAP}})^{\mathsf{T}}\mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{MAP}}), \end{aligned} \tag{6.4}$$

where the second term in the first line is canceled by using Eq. (6.3). Equation (6.4) is a basic equation in this chapter. $\mathbf{H}$ is a $J \times J$ Hessian matrix defined as

$$\mathbf{H} \triangleq \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{MAP}}}. \tag{6.5}$$

By using Eq. (6.1), it can also be represented as

$$\mathbf{H} \triangleq -\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathbf{O})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{MAP}}}. \tag{6.6}$$

The Hessian matrix also appeared in Eq. (5.9), as the second derivative of the negative log posterior. The determinant of this matrix plays an important role in the Occam factor, which penalizes the model complexity. Note that the Hessian matrix is a symmetric matrix based on the following property:

$$\begin{aligned} [\mathbf{H}]_{ij} &= \frac{\partial}{\partial \theta_i}\frac{\partial}{\partial \theta_j} f(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{MAP}}} \\ &= \frac{\partial}{\partial \theta_j}\frac{\partial}{\partial \theta_i} f(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\text{MAP}}} = [\mathbf{H}]_{ji}. \end{aligned} \tag{6.7}$$

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

Thus, by substituting the Taylor expansion of $f(\boldsymbol{\theta})$ (Eq. (6.4)) into Eq. (6.1), we obtain the approximated form of $p(\boldsymbol{\theta}|\mathbf{O})$, as follows:

$$\begin{aligned} p(\boldsymbol{\theta}|\mathbf{O}) &= \exp\left(-f(\boldsymbol{\theta})\right) \\ &\approx \exp\left(-f(\boldsymbol{\theta}^{\mathrm{MAP}}) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{MAP}})^{\intercal}\mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{MAP}})\right) \\ &\propto \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{MAP}})^{\intercal}\mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{MAP}})\right). \end{aligned} \tag{6.8}$$

Here, $\exp\left(-f(\boldsymbol{\theta}^{\mathrm{MAP}})\right)$ does not depend on θ, and we can neglect it. Thus, $p(\boldsymbol{\theta}|\mathbf{O})$ is approximated by the following Gaussian distribution with mean vector $\boldsymbol{\theta}^{\mathrm{MAP}}$ and covariance matrix $\mathbf{H}^{-1}$:

$$p(\boldsymbol{\theta}|\mathbf{O}) \approx \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}^{\mathrm{MAP}}, \mathbf{H}^{-1}). \tag{6.9}$$

Note that since the dimensionality of the Hessian matrix $\mathbf{H}$ is the number of parameters J, and it is often large in our practical problems, we would have a numerical issue as to how to obtain $\mathbf{H}^{-1}$. To summarize the Laplace approximation, if we have an arbitrary distribution $p(\boldsymbol{\theta}|\mathbf{O})$ with given continuous distribution $f(\boldsymbol{\theta})$, the Laplace approximation provides an analytical Gaussian distribution by using the mode $\boldsymbol{\theta}^{\mathrm{MAP}}$ and the Hessian matrix $\mathbf{H}$. $\boldsymbol{\theta}^{\mathrm{MAP}}$ is usually obtained by some numerical computation or the MAP estimation.

Now, let us consider the relationship between MAP (Chapter 4) and Laplace approximations. As we discussed in Eq. (4.9), the MAP approximation of the posterior distribution is represented by a Dirac delta function as follows:

$$p^{\mathrm{MAP}}(\boldsymbol{\theta}|\mathbf{O}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{MAP}}). \tag{6.10}$$

Therefore, Eq. (6.9) approaches the MAP solution, when the variance in the Gaussian distribution in Eq. (6.9) becomes very small. This often arises when the amount of training data is very large. In this sense, the Laplace approximation is a more precise approximation for the true posterior distribution than MAP, in terms of the asymptotic approximation. This precision of the Laplace approximation comes from consideration of the covariance matrix (Hessian matrix $\mathbf{H}$) effect in Eq. (6.9).

The Laplace approximation is widely used as an approximated Bayesian inference method for various topics (e.g., Bayesian logistic regression (Spiegelhalter & Lauritzen 1990, Genkin, Lewis & Madigan 2007), Gaussian processes (Rasmussen & Williams 2006), and Bayesian neural networks (MacKay 1992*c*)). Sections 6.3 and 6.4 discuss the applications of the Laplace approximation to Bayesian predictive classification and neural networks in acoustic modeling, respectively.

Note that the Laplace approximation is usually used to obtain the posterior distribution for non-exponential family distributions. Although the Laplace approximation can be used to deal with latent variable problems in the posterior distribution (e.g., Gaussian mixture model), the assumption of approximating a multiple peak distribution with a

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

single peak Gaussian is not adequate in many cases. Therefore, the Laplace approximation is often used with the other approximations based on the EM algorithm (MAP–EM, VB–EM) or sampling algorithm (MCMC) that handle latent variable problems.

6.2 Bayesian information criterion

This section describes the Bayesian information criterion in general. We focus on the posterior distribution of model $p(M|\mathbf{O})$, which is introduced in Section 3.8.7. We first think about a simple case when there is no latent variable. Then, $p(M|\mathbf{O})$ is represented by using the Bayes rule and the sum rule as follows:

$$\begin{aligned} p(M|\mathbf{O}) &= \frac{p(\mathbf{O}|M)p(M)}{p(\mathbf{O})} \\ &= \frac{\int p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)d\boldsymbol{\theta}p(M)}{p(\mathbf{O})}, \end{aligned} \tag{6.11}$$

where $p(M)$ denotes a prior distribution for a model M. Similarly to the formalization for Laplace approximation, we use a vector form of model parameters, that is $\boldsymbol{\theta} \in \mathbb{R}^J$, and J is the number of parameters. Then, $p(\boldsymbol{\theta}|M)$ denotes a prior distribution for a model parameter $\boldsymbol{\theta}$ given M, and $p(\mathbf{O}|\boldsymbol{\theta}, M)$ denotes a likelihood function given $\boldsymbol{\theta}$ and M.

Suppose we want to compare two types of models (M_1 and M_2) in terms of the posterior values of these models. Figure 6.1 compares fitting the same data with a single Gaussian (M_1) or two Gaussians (M_2). For this comparison, we can compute the following ratio of $p(M_1|\mathbf{O})$ and $p(M_2|\mathbf{O})$ and check whether the ratio is larger/less than 1:

$$\begin{aligned} \frac{p(M_1|\mathbf{O})}{p(M_2|\mathbf{O})} &= \frac{\frac{\int p(\mathbf{O}|\boldsymbol{\theta}_1, M_1)p(\boldsymbol{\theta}_1|M_1)d\boldsymbol{\theta}_1 p(M_1)}{p(\mathbf{O})}}{\frac{\int p(\mathbf{O}|\boldsymbol{\theta}_2, M_2)p(\boldsymbol{\theta}_2|M_2)d\boldsymbol{\theta}_2 p(M_2)}{p(\mathbf{O})}} \\ &= \frac{\int p(\mathbf{O}|\boldsymbol{\theta}_1, M_1)p(\boldsymbol{\theta}_1|M_1)d\boldsymbol{\theta}_1 p(M_1)}{\int p(\mathbf{O}|\boldsymbol{\theta}_2, M_2)p(\boldsymbol{\theta}_2|M_2)d\boldsymbol{\theta}_2 p(M_2)}. \end{aligned} \tag{6.12}$$

This factor is called the *Bayes factor* (Kass & Raftery 1993, 1995). To obtain the Bayes factor, we need to compute the marginal likelihood term:

$$\int p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)d\boldsymbol{\theta}. \tag{6.13}$$

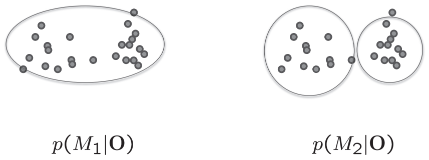

Figure 6.1 An example of model comparison, fitting the same data with a single Gaussian (M_1) or two Gaussians (M_2). By comparing the posterior probabilities of $p(M_1|\mathbf{O})$ and $p(M_2|\mathbf{O})$, we can select a more probable model.

Since it is difficult to solve this integral analytically, the following section approximates this calculation based on the asymptotic approximation, similarly to the Laplace approximation, as discussed in Section 6.1.

We first define the following continuous function g over $\boldsymbol{\theta}$:

$$p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M) \triangleq \exp\left(-g(\boldsymbol{\theta})\right). \tag{6.14}$$

This is slightly different from the definition of function f in Eq. (6.1), as Eq. (6.1) focuses on the posterior distribution of the parameters. Then, similarly to the previous section, we define the following MAP value:

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M) \\ &= \arg\min_{\boldsymbol{\theta}} g(\boldsymbol{\theta}).\end{aligned} \tag{6.15}$$

By using the Taylor expansion result in Eq. (6.4) with $\boldsymbol{\theta}^*$, Eq. (6.13) can be represented as follows:

$$\begin{aligned}&\int p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)d\boldsymbol{\theta} \\ &\quad\approx \exp\left(-g(\boldsymbol{\theta}^*)\right)\int \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^{\mathsf{T}}\mathbf{H}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)\right)d\boldsymbol{\theta} \\ &\quad= \exp\left(-g(\boldsymbol{\theta}^*)\right)\frac{(2\pi)^{\frac{J}{2}}}{|\mathbf{H}|^{\frac{1}{2}}},\end{aligned} \tag{6.16}$$

where J is the number of dimensions of $\boldsymbol{\theta}$, that is the number of parameters. The Hessian matrix $\mathbf{H}$ is defined with $g(\boldsymbol{\theta})$ as

$$\mathbf{H} \triangleq \nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}. \tag{6.17}$$

Or it is defined with $p(\mathbf{O}|\boldsymbol{\theta}, M)$ and $p(\boldsymbol{\theta}|M)$ as

$$\mathbf{H} \triangleq -\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} \log\left(p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)\right)|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}. \tag{6.18}$$

The integral in Eq. (6.16) can be solved by using the following normalization property of a Gaussian distribution:

$$\begin{aligned}&\int \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}^*, \mathbf{H}^{-1})d\boldsymbol{\theta} = \frac{|\mathbf{H}|^{\frac{1}{2}}}{(2\pi)^{\frac{J}{2}}}\int \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^{\mathsf{T}}\mathbf{H}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)\right)d\boldsymbol{\theta} = 1 \\ &\Longrightarrow \int \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)^{\mathsf{T}}\mathbf{H}(\boldsymbol{\theta}-\boldsymbol{\theta}^*)\right)d\boldsymbol{\theta} = \frac{(2\pi)^{\frac{J}{2}}}{|\mathbf{H}|^{\frac{1}{2}}}.\end{aligned} \tag{6.19}$$

Thus, from Eq. (6.16), the marginal likelihood is computed by using the determinant of the Hessian matrix $|\mathbf{H}|^{\frac{1}{2}}$.

Here, we focus on the determinant of the Hessian matrix $|\mathbf{H}|$. From the definition in Eq. (6.18), the Hessian matrix can be represented as:

$$\begin{aligned}\mathbf{H} &= -\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} \log p(\mathbf{O}|\boldsymbol{\theta}, M)|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} - \nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|M)|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} \\ &\approx -\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}} \log p(\mathbf{O}|\boldsymbol{\theta}, M)|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}.\end{aligned} \tag{6.20}$$

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

Thus, the Hessian matrix is decomposed to the logarithmic likelihood and prior term, and we neglect the prior term that does not depend on the amount of data. If we assume that the observed data are independently and identically distributed (iid), the likelihood term of T frame feature vectors is approximated when T is very large as[1]

$$\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{O}|\boldsymbol{\theta}, M)|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} \approx -T\mathbf{I}, \tag{6.21}$$

where $\mathbf{I}$ is a Fisher information matrix obtained from a single observation, which does not depend on the amount of data. Therefore, we can approximate the determinant of the Hessian matrix as:

$$|\mathbf{H}|^{-\frac{1}{2}} \approx |T\mathbf{I}|^{-\frac{1}{2}} = T^{-\frac{J}{2}}|\mathbf{I}|. \tag{6.22}$$

This approximation implies that the Hessian matrix approaches zero when the amount of data is large. That is

$$\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}^*, \mathbf{H}^{-1}) \to \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \text{ when } T \to \infty. \tag{6.23}$$

Therefore, the Laplace approximation can be equivalent to the MAP approximation when the amount of data is large, which is a reasonable outcome.

In the BIC approximation, by substituting Eq. (6.22) into Eq. (6.16), we obtain

$$\begin{aligned} &\log\left(\int p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)\right) \\ &\quad \approx \log p(\mathbf{O}|\boldsymbol{\theta}^*, M) + \log p(\boldsymbol{\theta}^*|M) + \frac{J}{2}\log(2\pi) - \frac{J}{2}\log T - \frac{J}{2}\log|\mathbf{I}|. \end{aligned} \tag{6.24}$$

Further, by neglecting the terms that do not depend on T, we finally obtain the following simple equation:

$$\begin{aligned} \log p(M|\mathbf{O}) &\propto \log\left(\int p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)\right) \\ &\approx \underbrace{\log p(\mathbf{O}|\boldsymbol{\theta}^*, M)}_{\text{log likelihood}} - \underbrace{\frac{J}{2}\log T}_{\text{penalty term}} . \end{aligned} \tag{6.25}$$

Here, $\boldsymbol{\theta}^*$ is often approximated by the ML estimate $\boldsymbol{\theta}^{\mathrm{ML}}$ instead of $\boldsymbol{\theta}^{\mathrm{MAP}}$, since $\boldsymbol{\theta}^{\mathrm{MAP}}$ approaches $\boldsymbol{\theta}^{\mathrm{ML}}$ when $T \to \infty$, as discussed in Section 4.3.7. Model selection based on this equation is said to be based on the *Bayesian information criterion* (BIC), or Schwartz information criterion, which approximates the logarithmic function of the posterior distribution of a model $p(M|\mathbf{O})$. The first term on the right-hand-side is a log-likelihood term with the MAP estimate (it is often approximated by the ML estimate) of model parameter $\boldsymbol{\theta}$. It is well known that this log-likelihood value is always increased when the number of parameters increases. Therefore, the log-likelihood value from the maximum likelihood criterion cannot be used for model selection since it will always prefer the model that has the largest number of parameters. On the other hand, Eq. (6.25), which is based on the Bayesian criterion, has an additional term, which

[1] This proof is not obvious. See Ghosh, Delampady & Samanta (2007) for more detailed derivations.

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

is proportional to the number of model parameters, denoted by J, and the logarithmic number of observation frames, denoted by $\log T$. This term provides the penalty so that the total value in Eq. (6.25) is penalized not to select a model with too many parameters.

Comparing Eq. (6.25) with the regularized form of the objective function in Eq. (4.3) in the MAP estimation, Eq. (6.25) can be viewed as having an l^0 regularization term:

$$\begin{aligned}\log p(M|\mathbf{O}) &\approx \log p(\mathbf{O}|\boldsymbol{\theta}^*, M) - \frac{J}{2}\log T \\ &= \underbrace{\log p(\mathbf{O}|\boldsymbol{\theta}^*, M)}_{\text{log likelihood}} - \underbrace{\frac{\|\boldsymbol{\theta}\|_0}{2}\log T}_{l^0 \text{ regularization term}} .\end{aligned} \tag{6.26}$$

Therefore, the BIC objective function can also be discussed within a regularization perspective. Another well-known model selection criterion, the Akaike information criterion (AIC) (Akaike 1974), is similarly represented by

$$\begin{aligned}-\frac{1}{2}\text{AIC} &= \log p(\mathbf{O}|\boldsymbol{\theta}^*, M) - J \\ &= \log p(\mathbf{O}|\boldsymbol{\theta}^*, M) - \|\boldsymbol{\theta}\|_0.\end{aligned} \tag{6.27}$$

It also has the l^0 regularization term, but it does not depend on the amount of data T.

Using the BIC, we can simplify the model comparison based on the Bayes factor. That is, by substituting Eq. (6.25) into the logarithm of the Bayes factor (Eq. (6.12)) we can also obtain the following equation:

$$\log\left(\frac{p(M_1|\mathbf{O})}{p(M_2|\mathbf{O})}\right) \approx \log p(\mathbf{O}|\boldsymbol{\theta}_1^*, M_1) - \log p(\mathbf{O}|\boldsymbol{\theta}_2^*, M_2) - \frac{J_1 - J2}{2}\log T. \tag{6.28}$$

Therefore, we can perform the model comparison by checking the sign of Eq. (6.28). If the sign is positive, M_1 is a more appropriate model in the BIC sense.

However, since we usually cannot assume that a single Gaussian distribution applies to a complicated model and/or that there are enough data to satisfy the large sample approximation, the BIC assumption is not valid for our practical problems. Therefore, in practical use, we introduce a tuning parameter λ which heuristically controls the balance of the log-likelihood and penalty terms, as follows:

$$\log\left(\int p(\mathbf{O}|\boldsymbol{\theta}, M)p(\boldsymbol{\theta}|M)\right) \approx \log p(\mathbf{O}|\boldsymbol{\theta}^*, M) - \lambda\frac{J}{2}\log T. \tag{6.29}$$

This actually works effectively for the model selection problems of HMMs and GMMs, including speech processing (Chou & Reichl 1999, Shinoda & Watanabe 2000) and image processing (Stenger, Ramesh, Paragios *et al.* 2001). We discuss the applications of BIC to speech processing in Sections 6.5 and 6.6.

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

6.3 Bayesian predictive classification

This section addresses how the Laplace approximation is developed to build the Bayesian predictive classification (BPC) approach which has been successfully applied for robust speech recognition in noisy environments (Jiang *et al.* 1999, Huo & Lee 2000, Lee & Huo 2000, Chien & Liao 2001). As mentioned in Section 3.1, the Bayes decision theory for speech recognition can be realized as the BPC rule where the expected loss function in Eq. (3.16) is calculated by marginalizing over the continuous density HMM parameters $\Theta = \{\boldsymbol{\pi} = \{\pi_j\}, A = \{a_{ij}\}, B = \{\omega_{jk}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}\}\}$, as discussed in Section 3.2.3, as well as the *n*-gram parameters $\Theta = \{p(w_i|w_{i-n+1}^{i-1})\}$, as discussed in Section 3.6. Here, we only focus on the acoustic models based on HMMs. The marginalization over all possible parameter values aims to construct a decision rule which considers the *prior uncertainty* of HMM parameters. The resulting speech recognition is robust to mismatch conditions between training and test data. In general, such marginalization could compensate some other ill-posed conditions due to the uncertainties and variations from the observed data and the assumed models. In this section, we first describe how a Bayesian decision rule is established for robust speech recognition and how the uncertainties of model parameters are characterized. Then, the approximate solution to the HMM-based decision rule is formulated by using the Laplace approximation. We also present some other extensions of the BPC decision rule.

6.3.1 Robust decision rule

In automatic speech recognition, we usually apply the plug-in maximum a-posteriori (MAP) decision rule, as given in Eq. (3.2), and combine the approximate acoustic model $\hat{p}_\Theta(\mathbf{O}|W)$ and language model $\hat{p}_\Theta(W)$ for finding the most likely word sequence W corresponding to an input speech sentence $\mathbf{O} = \{\mathbf{o}_t\}$. The maximum likelihood (ML) parameters Θ^{ML}, as described in Section 3.4, and the MAP parameters Θ^{MAP}, as formulated in Section 4.3, are treated as the point estimates and plugged into the MAP decision for speech recognition. However, in practical situations, speech recognition systems are occasionally applied in noisy environments. The training data may be collected in ill-posed conditions, where the HMM parameters may be over-trained with limited training data conditions, and can lose the generalization capability for future data conditions. Thus, the assumed model may not properly represent the real-world speech data. For this situation, the plug-in MAP decision, relying only on the single best HMM parameter values Θ^{ML} or Θ^{MAP}, is risky for speech recognition. Because of these considerations, we are motivated to establish a robust decision rule which accommodates the uncertainties of the assumed model and the collected data so that the error risks in speech recognition can be reduced.

In the HMM framework (Section 3.2), the prior uncertainty of GMM parameters $\Theta_j = \{\omega_{jk}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}|k = 1, \cdots, K\}$ in an HMM state j should be characterized from training data and then applied for a BPC decision of future data. In this section, we use the following notation to clearly distinguish training and future data:

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

$$\begin{aligned}\mathbf{O} &: \text{future data,}\\ \mathcal{O} &: \text{training data.}\end{aligned} \tag{6.30}$$

Let the distribution $p(\mathbf{o}_t|\Theta_j)$ of a future observation frame $\mathbf{o}_t$ under an HMM state j be distorted in an admissible set $\mathcal{P}_j(\epsilon_j)$ with a distortion level $\epsilon_j \leq 0$ which is denoted by

$$\mathcal{P}_j(\epsilon_j) = \{p(\mathbf{o}_t|\Theta_j)|\Theta_j \in \Omega(\epsilon_j)\}, \tag{6.31}$$

where $\Omega(\epsilon_j)$ denotes the admission region of the GMM parameter space. In the special case of no distortion ($\epsilon_j = 0$), $\mathcal{P}_j(0) = \{p(\mathbf{o}_t|\Theta_j^{(0)})\}$ is a singleton set which consists of the ideal and non-distorted distribution of an HMM state j with the parameters $\Theta_j^{(0)}$ estimated from a training set $\mathcal{O}$. However, in real-world applications, there exist many kinds of distortions ($\epsilon_j > 0$) between the trained models and the test observations. These distortions and variations should be modeled and incorporated in the Bayes decision to achieve robustness of speech recognition in adverse environments.

The Bayesian inference approach provides a good way to formalize this parameter uncertainty problem and formulate the solution to the robust decision rule. To do so, we intend to consider the uncertainty of the HMM parameters Θ to be random. An a-priori distribution $p(\Theta|\Psi)$ could serve as the prior knowledge about Θ, where $\Theta \in \Omega$ are located in a region of interest of the HMM parameter space Ω, and Ψ denotes the corresponding hyperparameters. Such prior information may come from subject considerations and/or from previous experience. Therefore, the BPC decision rule is established as

$$\begin{aligned}\hat{W} = d_{\text{BPC}}(\mathbf{O}) &= \arg\max_W \tilde{p}(W|\mathbf{O})\\ &= \arg\max_W \tilde{p}_\Theta(\mathbf{O}|W)p_\Theta(W),\end{aligned} \tag{6.32}$$

where

$$\tilde{p}(\mathbf{O}|W) = \int_\Omega p(\mathbf{O}|\Theta, W)p(\Theta|\mathcal{O}, W)d\Theta. \tag{6.33}$$

Here, $p(\Theta|\mathcal{O}, W)$ denotes the posterior distribution of HMM parameters Θ given the training utterances $\mathcal{O}$, which is written as

$$p(\Theta|\mathcal{O}, W) = \frac{p(\mathcal{O}|\Theta, W)p(\Theta)}{\int_\Omega p(\mathcal{O}|\Theta, W)p(\Theta)d\Theta}. \tag{6.34}$$

In Eq. (6.33), the marginalization is performed over all possible HMM parameter values Θ. This marginalized distribution is also called the *predictive distribution*. Model uncertainties are considered in the BPC decision rule. The optimum BPC decision rule was illustrated in Nadas (1985).

The crucial difference between the plug-in MAP decision and the BPC decision is that the former acts as if the estimated HMM parameters were the true ones, whereas the latter averages over the uncertainty of parameters. In an extreme case, if $p(\Theta|\mathcal{O}, W) = \delta(\Theta - \Theta^{\text{ML/MAP}})$ with $\delta(\cdot)$ denoting the Dirac delta function with the ML or MAP estimate, the BPC decision rule coincides with the plug-in MAP decision

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

rule based on ML HMM parameters $\Theta^{\mathrm{ML/MAP}}$, i.e., the predictive distribution can be approximated as:

$$\begin{aligned}\tilde{p}(\mathbf{O}|W) &= \int_{\Omega} p(\mathbf{O}|\Theta, W)p(\Theta|\mathcal{O}, W)d\Theta \\ &\approx \int_{\Omega} p(\mathbf{O}|\Theta, W)\delta(\Theta - \Theta^{\mathrm{ML/MAP}})d\Theta \\ &= p(\mathbf{O}|\Theta^{\mathrm{ML/MAP}}, W). \end{aligned} \tag{6.35}$$

Therefore, the plug-in MAP decision rule with the ML/MAP estimate corresponds to an approximation of the BPC decision rule.

Using the continuous density HMMs, the missing data problem happens so that the predictive distribution is calculated by considering all sequences of HMM states $S = \{s_t\}$ and mixture components $V = \{v_t\}$:

$$\begin{aligned}\tilde{p}(\mathbf{O}|W) &= \int p(\mathbf{O}|\Theta, W)p(\Theta|\mathcal{O}, W)d\Theta \\ &= \int p(\mathbf{O}|\Theta, W)p(\Theta|\Psi, W)d\Theta \\ &= \sum_{S,V} \int p(\mathbf{O}, S, V|\Theta, W)p(\Theta|\Psi, W)d\Theta \\ &= \sum_{S,V} \int p(\mathbf{O}|S, V, \Theta, W)p(S, V, |\Theta, W)p(\Theta|\Psi, W)d\Theta \\ &\triangleq \mathbb{E}_{(S,V,\Theta)}[p(\mathbf{O}|S, V, \Theta, W)], \end{aligned} \tag{6.36}$$

where the posterior distribution $p(\Theta|\mathcal{O}, W)$ given training data $\mathcal{O}$ is replaced by the prior distribution $p(\Theta|\Psi, W)$ given hyperparameters Ψ in accordance with an empirical Bayes theory. This predictive distribution is seen as an integration of likelihood function $p(\mathbf{O}|S, V, \Theta, W)$ over all possible values of $\{S, V, \Theta\}$. However, the computation of predictive distribution over all state sequences S and mixture component sequences V is complicated and expensive. A popular way to deal with this computation is to employ the Viterbi approximation, as we discussed in Section 3.3.2, to compute the approximate predictive distribution (Jiang *et al.* 1999):

$$\tilde{p}(\mathbf{O}|W) \approx \max_{S,V} \int p(\mathbf{O}, S, V|\Theta, W)p(\Theta|\Psi, W)d\Theta. \tag{6.37}$$

Therefore, the approximated predictive distribution only considers marginalization over the model parameter Θ. The following sections describe how to deal with model parameter marginalization.

6.3.2 Laplace approximation for BPC decision

In this section, Laplace approximation is adopted to approximate the integral in calculation of the predictive distribution $\tilde{p}(\mathbf{O}, S, V|W)$ in Eq. (6.36) or in Eq. (6.37), given state

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

and mixture component sequences S and V. Let us define the continuous function $g(\Theta)$, similarly to Eqs. (6.1) and (6.14):

$$p(\mathbf{O}, S, V|\Theta, W)p(\Theta|\Psi, W) = \exp(-g(\Theta)). \tag{6.38}$$

The value of Θ that minimizes $g(\Theta)$ is obtained as a mode or an MAP estimate of HMM parameters,

$$\begin{aligned}\Theta^{\mathrm{MAP}} &= \arg\min_{\Theta} \; g(\Theta) \\ &= \arg\max_{\Theta} \; p(\mathbf{O}, S, V|\Theta, W)p(\Theta|\Psi, W).\end{aligned} \tag{6.39}$$

Note that we have a solution to the above optimization based on the MAP estimation, as we discussed in Section 4.3. Then since we have the equation $\nabla_{\Theta} g(\Theta)|_{\Theta=\Theta^{\mathrm{MAP}}} = 0$ from Eq. (6.39), a second-order Taylor expansion around Θ^{MAP} can be used to approximate $g(\Theta)$ as follows:

$$g(\Theta) \approx g(\Theta^{\mathrm{MAP}}) + \frac{1}{2}(\Theta - \Theta^{\mathrm{MAP}})^{\mathsf{T}}\mathbf{H}(\Theta - \Theta^{\mathrm{MAP}}), \tag{6.40}$$

where $\mathbf{H}$ denotes a Hessian matrix defined as

$$\mathbf{H} \triangleq \nabla_{\Theta}\nabla_{\Theta} g(\Theta)|_{\Theta=\Theta^{\mathrm{MAP}}}. \tag{6.41}$$

We should note that we use a vector representation of Θ (i.e., $\Theta \in \mathbb{R}^M$ and M is the number of all CDHMM parameters) by arranging all CDHMM parameters $(\{a_{ij}, \omega_{jk}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}\})$ to form a huge vector, to make the above gradient well defined.[2]

Therefore, we obtain the approximate predictive distribution, which is derived from

$$\begin{aligned}\tilde{p}(\mathbf{O}, S, V|W) &= \int \exp(-g(\Theta))d\Theta \\ &\approx \exp(-g(\Theta^{\mathrm{MAP}})) \int \exp\left(-\frac{1}{2}(\Theta - \Theta^{\mathrm{MAP}})^{\mathsf{T}}\mathbf{H}(\Theta - \Theta^{\mathrm{MAP}})\right)d\Theta.\end{aligned} \tag{6.42}$$

Similarly to Eq. (6.18), the predictive distribution in Eq. (6.42) is approximately obtained through arranging a multivariate Gaussian distribution in the integrand of Eq. (6.42) as follows:

$$\begin{aligned}\tilde{p}(\mathbf{O}, S, V|W) \approx\; & p(\mathbf{O}, S, V|\Theta^{\mathrm{MAP}}, W) \\ & \times p(\Theta^{\mathrm{MAP}}|\Psi, W)(2\pi)^{M/2}|\mathbf{H}|^{-1/2}.\end{aligned} \tag{6.43}$$

As a result, the BPC decision based on Laplace approximation is implemented by Eq. (6.43). The logarithm of predictive distribution, $\log \tilde{p}(\mathbf{O}, S, V|W)$, is seen as follows:

$$\begin{aligned}&\log \tilde{p}(\mathbf{O}, S, V|W) \\ &\quad\approx \log\left(p(\mathbf{O}, S, V|\Theta^{\mathrm{MAP}}, W)p(\Theta^{\mathrm{MAP}}|\Psi, W)\right) + \log\left((2\pi)^{M/2}|\mathbf{H}|^{-1/2}\right).\end{aligned} \tag{6.44}$$

[2] We should also note that most CDHMM parameters have some constraint (e.g., state transitions and mixture weights have positivity and sum-to-one constraint) and the gradient of these parameters should consider these constraints. However, the following example only considers mean vector parameters, and we do not have this constraint problem.

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

The likelihood of the first term can be obtained by using the MAP–EM (or Viterbi approximation), as we discussed in Section 4.3. The second term is based on the logarithm of the determinant of the Hessian matrix. From Eq. (6.9), the Hessian matrix can be interpreted as the precision (inverse covariance) matrix of the model parameters ($p(\Theta|\mathbf{O}) \approx \mathcal{N}(\Theta|\Theta^{\mathrm{MAP}}, \mathbf{H}^{-1})$). Therefore, if the model parameters are uncertain, the determinant of the covariance matrix of Θ becomes large (i.e., the determinant of the Hessian matrix $\mathbf{H}$ becomes small), the log likelihood of the predictive distribution is decreased. Thus, considering model uncertainty in the BPC decision is meaningfully reflected in Eq. (6.43).

6.3.3 BPC decision considering uncertainty of HMM means

To simplify the discussion, we only consider the uncertainty of the mean vectors in HMM parameters for BPC decoding, as the mean vectors are the most dominant parameters for the ASR performance. In addition, by only focusing on the mean vectors, we can avoid the difficulty of applying the Laplace approximation to the covariance and weight parameters, which have constraints, assuming the HMM means of state j[3] and mixture component k in dimension d are independently generated from Gaussian prior distributions with the Hessian matrix as the precision matrix.

The joint prior distribution $p(\Theta|\Psi, W)$ is expressed by

$$p(\{\mu_{jkd}\}|\{\mu^0_{jkd}, \Sigma^0_{jkd}\}, W) = \prod_{j=1}^{J}\prod_{k=1}^{K}\prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\,\Sigma^0_{jkd}}} \exp\left[-\frac{(\mu_{jkd} - \mu^0_{jkd})^2}{2\Sigma^0_{jkd}}\right], \tag{6.45}$$

with a collection of hyperparameters including Gaussian mean vectors $\boldsymbol{\mu}^0_{jk} = \{\mu^0_{jkd}\}$ and diagonal covariance matrices $\boldsymbol{\Sigma}^0_{jk} = \mathrm{diag}\,\{\Sigma^0_{jkd}|d = 1, \cdots, D\}$ for different states and mixture components. The GMM mixture weights $\{\omega_{jk}\}$ and diagonal covariance matrices $\{\boldsymbol{\Sigma}_{jk} = \mathrm{diag}\,\{\Sigma_{jkd}|d = 1, \cdots, D\}\}$ of the original CDHMM are assumed to be deterministic without uncertainty. Note that the above prior distribution is conditional on the word sequence W, as it is required in the BPC decision rule in Eq. (6.36). This condition means that we only deal with the prior distributions of HMM parameters appearing in the state sequences obtained by hypothesized word sequence W. But this condition can be disregarded since this condition for the prior distribution does not matter in the following analysis.

Given an unknown test utterance $\mathbf{O} = \{\mathbf{o}_t|t = 1, \cdots, T\}$ and the unobserved state sequence $S = \{s_t|t = 1, \cdots, T\}$ and mixture component sequence $V = \{v_t|t = 1, \cdots, T\}$ obtained by hypothesized word sequence W, we combine the Gaussian likelihood function $p(\mathbf{O}|S, V, \{\mu'_{jkd}\}, W)$ and the Gaussian prior distribution $p(\{\mu'_{jkd}\}|\{\mu^0_{jkd}, \Sigma^0_{jkd}\}, W)$ based on the MAP–EM approach (Section 4.2) to give:

[3] Note that index j denotes all HMM states for all phonemes, unlike index j for a phoneme in Section 3.2. Therefore, the number of HMM states J appearing later in this section also means the number of all HMM states used in the acoustic models, which could be several thousand when we use a standard LVCSR setup.

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

$$\begin{aligned}\log p(\{\mu'_{jkd}\}|\mathbf{O}, W) &\approx Q^{\mathrm{MAP}}(\{\mu'_{jkd}\}|\{\mu_{jkd}\}) \\ &= \mathbb{E}_{(S,V)}[\log p(\mathbf{O}, S, V|\{\mu'_{jkd}\}, W)|\mathbf{O}, \{\mu_{jkd}\}] \\ &\quad + \log p(\{\mu'_{jkd}\}|\{\mu^0_{jkd}, \Sigma^0_{jkd}\}, W).\end{aligned} \tag{6.46}$$

Equation (6.46) is further rewritten to come up with an approximate posterior distribution which is arranged as a Gaussian distribution:

$$\begin{aligned}\log p(\{\mu'_{jkd}\}|\mathbf{O}, W) &\approx \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} \gamma_t(j,k)\Big[\log p(\mathbf{o}_t|s_t = j, v_t = k, \mu'_{jkd}, W) \\ &\qquad + \log p(\mu'_{jkd}|\mu^0_{jkd}, \Sigma^0_{jkd}, W)\Big] \\ &= \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} \log \mathcal{N}(\mu'_{jkd}|\hat{\mu}_{jkd}, \hat{\Sigma}_{jkd}),\end{aligned} \tag{6.47}$$

with the hyperparameters derived as

$$\hat{\mu}_{jkd} = \frac{\Sigma_{jkd}}{c_{jk}\Sigma^0_{jkd} + \Sigma_{jkd}}\mu^0_{jkd} + \frac{c_{jk}\Sigma^0_{jkd}}{c_{jk}\Sigma^0_{jkd} + \Sigma_{jkd}}\bar{o}_{jkd}, \tag{6.48}$$

$$\frac{1}{\hat{\Sigma}_{jkd}} = \frac{1}{\Sigma^0_{jkd}} + \frac{c_{jk}}{\Sigma_{jkd}}, \tag{6.49}$$

where

$$\gamma_t(j,k) = p(s_t = j, v_t = k|\mathbf{O}, \{\mu_{jkd}\}, W), \tag{6.50}$$

$$c_{jk} = \sum_{t=1}^{T}\gamma_t(j,k), \tag{6.51}$$

$$\bar{\mathbf{o}}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)\mathbf{o}_t}{c_{jk}}. \tag{6.52}$$

The posterior distribution in Eq. (6.46) is obtained because we adopt the conjugate prior to model the uncertainty of HMM means. Recall the MAP estimation result of the mean vector in Eq. (6.53):

$$\hat{\mu}_{jkd} \triangleq \frac{\phi^{\mu}_{jkd}\mu^0_{jkd} + \sum_{t=1}^{T}\gamma_t(j,k)o_{td}}{\phi^{\mu}_{jk} + \sum_{t=1}^{T}\gamma_t(j,k)}. \tag{6.53}$$

Comparing Eqs. (6.53) and (6.48), it is apparent that Eq. (6.48) based on the BPC solution considers the effect of Σ^0_{jkd}.

An EM algorithm is iteratively applied to approximate the log posterior distribution of new estimate $\{\mu'_{jkd}\}$, $\log p(\{\mu'_{jkd}\}|\mathbf{O}, W)$, by using the posterior auxiliary function of new estimate $\{\mu'_{jkd}\}$ given the current estimate $\{\mu_{jkd}\}$, $Q^{\mathrm{MAP}}(\{\mu'_{jkd}\}|\{\mu_{jkd}\})$. The calculation converges after several EM iterations. Given the updated occupation probability, the Gaussian posterior distribution of $\{\mu'_{jkd}\} = [\boldsymbol{\mu}'^{\mathsf{T}}_{11}, \boldsymbol{\mu}'^{\mathsf{T}}_{12}, \cdots, \boldsymbol{\mu}'^{\mathsf{T}}_{JK}]^{\mathsf{T}}$, which

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

concatenates all Gaussian mean vectors, is calculated and used to find the approximation in Eq. (6.9):

$$\begin{aligned} p(\{\mu'_{jkd}\}|\mathbf{O}, W) &\approx \mathcal{N}(\{\mu'_{jkd}\}|\{\hat{\mu}_{jkd}\}, \{\hat{\Sigma}_{jkd}\}) \\ &\approx \mathcal{N}(\{\mu'_{jkd}\}|\{\mu^{\text{MAP}}_{jkd}\}, \{[(\mathbf{H}^{\mu})^{-1}]_{jkd}\}). \end{aligned} \tag{6.54}$$

Therefore, we substitute

$$\mu^{\text{MAP}}_{jkd} = \hat{\mu}_{jkd}, \tag{6.55}$$

$$[(\mathbf{H}^{\mu})^{-1}]_{jkd} = \hat{\Sigma}_{jkd}, \tag{6.56}$$

into Eq. (6.43). This approximation avoids a need to directly compute the Hessian matrix, where the number of dimensions of the Hessian matrix is very large ($J \times K \times D$). The BPC-based recognition is accordingly implemented for robust speech recognition.

In Chien & Liao (2001), the BPC decision was extended to a transformation-based BPC decision where the uncertainties of transformation parameters were taken into account. The transformation parameters of HMM means and variances were characterized by the Gaussian–Wishart distribution. Using this decision, HMM parameters were treated as deterministic values. In Chien (2003), the transformation-based BPC was further combined with the maximum likelihood linear regression adaptation (Leggetter & Woodland 1995). The linear regression based BPC was proposed for noisy speech recognition. The transformation regression parameters were represented by the multivariate Gaussian distribution. The predictive distributions in these two methods were approximated by calculating the predictive distributions for individual frames $\mathbf{o}_t$ without involving the Laplace approximation.

6.4 Neural network acoustic modeling

Acoustic modeling based on the *deep neural network* (DNN) is now a new trend towards achieving high performance in automatic speech recognition (Dahl, Yu, Deng *et al.* 2012, Hinton *et al.* 2012). The context-dependent DNNs were combined with HMMs and have recently shown significant improvements over discriminatively trained HMMs with state-dependent GMMs (Seide, Li, Chen *et al.* 2011). In particular, the DNN was proposed to conduct a greedy and layer-wise pretraining of the weight parameters with either a supervised or unsupervised criterion (Hinton, Osindero & Teh 2006, Salakhutdinov 2009). This pretraining step prevents the supervised training of the network from being trapped in a poor local optimum. In general, there are five to seven hidden layers with thousands of sigmoid non-linear neurons in a DNN model. The output layer consists of softmax non-linear neurons. In addition to the pretraining scheme, the success of the DNN acoustic model also comes from the tandem processing, frame randomization (Seide *et al.* 2011) and (Hessian-free) sequence training (Kingsbury, Sainath & Soltau 2012, Veselỳ, Ghoshal, Burget *et al.* 2013). Without loss of generality, we address the artificial neural network (NN) acoustic model for ASR, based on the feed-forward multilayer perceptron with a single hidden layer, as

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

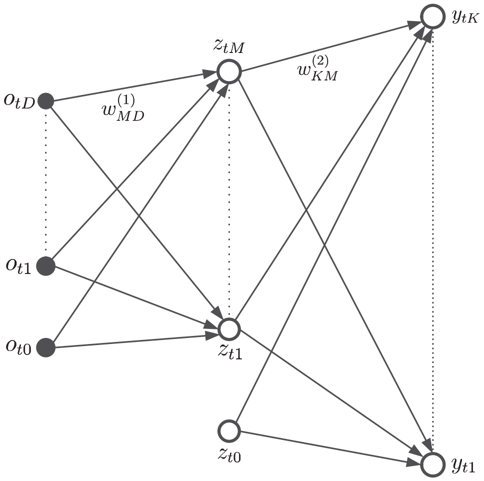

Figure 6.2 A neural network based on the multilayer perceptron with a single hidden layer. $\mathbf{o}_t \in \mathbb{R}^D$, $\mathbf{z}_t \in \mathbb{R}^M$, and $\mathbf{y}_t \in \mathbb{R}^K$ denote the input vector, hidden unit vector, and output vector at frame t, respectively. o_{t0} and z_{t0} are introduced to consider the bias terms in the transformation, and these are usually set with some fixed values (e.g., $o_{t0} = z_{t0} = 1$). $\mathbf{w}^{(1)} \in \mathbb{R}^{M(D+1)}$ and $\mathbf{w}^{(2)} \in \mathbb{R}^{K(M+1)}$ denote the weight vectors (to use a Laplace approximation, we represent these values as the vector representation rather than the matrix representation) from the input to hidden layers and from the hidden to output layers, respectively.

illustrated in Figure 6.2. We bring in the issue of model regularization in construction of NNs and present the *Bayesian neural networks* (MacKay 1992*c*, Bishop 2006) for robust speech recognition. This framework can obviously be extended to DNN-based speech recognition with many hidden layers. Importantly, we introduce a prior distribution to express the uncertainty of synaptic weights in Bayesian NNs. This uncertainty serves as a penalty factor to avoid too-complicated or over-trained models. The variations of acoustic models due to the heterogeneous training data could be compensated through Bayesian treatment. The Laplace approximation is applied to derive the predictive distribution for robust classification of a speech signal. In what follows, we first describe the combination of Bayesian NNs and HMMs for the application of speech recognition. MAP estimation is applied to find a solution to adaptive NN training. A Laplace approximation is developed to construct Bayesian NNs for HMM-based speech recognition.

6.4.1 Neural network modeling and learning

A standard neural network for a K-class classification problem is depicted in Figure 6.2, which is established as a feed-forward network function with a layer structure. This function maps the D input neurons $\mathbf{o}_t = \{o_{ti}\} \in \mathcal{R}^D$ to K output neurons $\mathbf{y}_t(\mathbf{o}_t, \mathbf{w}) = \{y_{tk}(\mathbf{o}_t, \mathbf{w})\} \in \mathcal{R}^K$ where

$$y_{tk}(\mathbf{o}_t, \mathbf{w}) = \frac{\exp\left(a_k(\mathbf{o}_t, \mathbf{w})\right)}{\sum_j \exp\left(a_j(\mathbf{o}_t, \mathbf{w})\right)} \tag{6.57}$$

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

is given by a softmax non-linear function, or the normalized exponential function which satisfies $0 \leq y_{tk} \leq 1$ and $\sum_{k=1}^{K} y_{tk} = 1$. In Eq. (6.57), $a_k(\mathbf{o}_t, \mathbf{w})$ is calculated as the output unit activation given by

$$a_k(\mathbf{o}_t, \mathbf{w}) = \sum_{j=0}^{M} w_{kj}^{(2)} \mathrm{Sigmoid}\left(\sum_{i=0}^{D} w_{ji}^{(1)} o_{ti}\right), \tag{6.58}$$

where

$$\mathrm{Sigmoid}(a) = \frac{1}{1+\exp(-a)} \tag{6.59}$$

denotes the logistic sigmoid function and

$$\mathbf{w} = \{w_{ji}^{(1)}, w_{kj}^{(2)} | 1 \leq i \leq D, 1 \leq j \leq M, 1 \leq k \leq K\} \tag{6.60}$$

denotes the synaptic weights of the first and the second layers, respectively. For model training, the target values of output neurons are assigned as Bernoulli variables where value 1 in the kth neuron means that $\mathbf{o}_t$ belongs to class k and the other neurons have target value 0. In this figure, $\phi_{t0} = z_{t0} = 1$ and the corresponding weights $\{w_{j0}^{(1)}, w_{k0}^{(2)}\}$ denote the bias parameters in neurons. This model is also known as the *multilayer perceptron*. Such a neural network model performs well for pattern recognition for two reasons. One is the nested non-linear function, which can learn the complicated deep structure from real-world data through feed-forwarding the input signals to the output classes layer by layer. The second reason is that it adopts the softmax function or the logistic sigmoid function Sigmoid$(\cdot)$ as the neuron processing unit, which is beneficial for building high-performance discriminative models. In general, deep neural networks are affordable for learning the invariance information and extracting the hierarchy of concepts or features from data. Higher-level concepts are defined from lower-level ones. The logistic sigmoid function and softmax function play an important role for binary classification and multi-class classification, respectively. These functions are essential for calculating posterior probability and establishing the discriminative model. Basically, we collect a set of training samples and their target values $\{\mathbf{o}_t, \mathbf{t}_t\}$. The synaptic weights are estimated by calculating an error function $E(\mathbf{w})$ and minimizing it through the gradient descent algorithm,

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}), \tag{6.61}$$

where η denotes the learning rate and τ denotes the iteration index. The sum-of-squares error function is calculated for the regression problem while the *cross entropy error function* is applied for the classification problem. The *error back-propagation* algorithm has been proposed to find gradient vector $\nabla E(\mathbf{w})$ and widely applied for training of individual synaptic weights $\mathbf{w} = \{w_{ji}^{(1)}, w_{kj}^{(2)}\}$ in different layers.

6.4.2 Bayesian neural networks and hidden Markov models

For the application of speech recognition, the NNs should be combined with the HMMs (denoted by NN–HMMs) to deal with *sequential training* of synaptic weights $\mathbf{w}$ from a sequence of speech feature vectors $\mathbf{O} = \{\mathbf{o}_t \in \mathbb{R}^D | t = 1, \cdots, T\}$. As discussed before,

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

this feature can be obtained by stacking the neighboring MFCC or filter bank features. For example, if the original features are represented by $\mathbf{O}^{\text{org}} = \{\mathbf{o}_t^{\text{org}} \in \mathbb{R}^{D^{\text{org}}} | t = 1, \cdots, T\}$, the stacked feature for the NN input is represented by

$$\mathbf{o}_t = [(\mathbf{o}_{t-L}^{\text{org}})^\mathsf{T}, \cdots, (\mathbf{o}_t^{\text{org}})^\mathsf{T}, \cdots, (\mathbf{o}_{t-L}^{\text{org}})^\mathsf{T}]^\mathsf{T}. \tag{6.62}$$

Then, the number of feature dimensions becomes $D = (2L + 1)D^{\text{org}}$. L is set between five and seven depending on the task. This expanded feature can model the short-range dynamics of speech features, as we model in the conventional GMM framework by using the delta cepstrum or linear discriminant analysis techniques (Furui 1986, Haeb-Umbach & Ney 1992).

However, the stacking technique cannot fully model the long-range speech dynamics. The Markov states are used to characterize the temporal correlation in sequential patterns. We estimate the NN–HMM parameters $\Theta = \{\boldsymbol{\pi}, A, \mathbf{w}\}$, consisting of initial state probabilities $\boldsymbol{\pi}$, state transition probabilities A and the N-dimensional synaptic weights $\mathbf{w}$, by maximizing the likelihood function $p(\mathbf{O}|\Theta)$. Each output neuron calculates the posterior probability of a context dependent HMM state k given a feature vector $\mathbf{o}_t$ using synaptic weights $\mathbf{w}$, i.e.,

$$y_k(\mathbf{o}_t, \mathbf{w}) \triangleq p(s_t = k|\mathbf{o}_t, \mathbf{w}). \tag{6.63}$$

ML estimation of Θ has an incomplete data problem and should be solved according to the EM algorithm. The EM algorithm is an iterative procedure where the E-step is to calculate an auxiliary function of training utterances $\mathbf{O}$ using new NN–HMM parameters Θ' given parameters Θ at the current iteration:

$$\begin{aligned} Q^{\text{ML}}(\Theta'|\Theta) &= \mathbb{E}_{(S)}\{\log p(\mathbf{O}, S|\Theta')|\mathbf{O}, \Theta\} \\ &= \sum_S p(S|\mathbf{O}, \Theta) \left[\sum_{t=1}^T (\log a'_{s_{t-1}s_t} + \log p(\mathbf{o}_t|s_t, \mathbf{w}')) \right] \\ &\propto \sum_{t=1}^T \sum_{k=1}^K \gamma_t(k) \Big[\log a'_{s_{t-1}s_t} + \log p(s_t = k|\mathbf{o}_t, \mathbf{w}') - \log p(s_t = k) \Big], \end{aligned} \tag{6.64}$$

where $\gamma_t(k) = p(s_t = k|\mathbf{O}, \Theta)$ denotes the state occupation probability. NN–HMM parameters Θ' at a new iteration are then estimated by M-step:

$$\mathbf{w}' = \arg\min_{\mathbf{w}'} \{-Q^{\text{ML}}(\mathbf{w}'|\mathbf{w})\}. \tag{6.65}$$

Here, we only consider the estimation of the synaptic weights $\mathbf{w}$ by minimizing the corresponding negative auxiliary function:

$$\begin{aligned} -Q^{\text{ML}}(\mathbf{w}'|\mathbf{w}) &\propto -\sum_{t=1}^T \sum_{k=1}^K \gamma_t(k) \log p(s_t = k|\mathbf{o}_t, \mathbf{w}) \\ &= -\sum_{t=1}^T \sum_{k=1}^K \gamma_t(k) \log y_k(\mathbf{o}_t, \mathbf{w}). \end{aligned} \tag{6.66}$$

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

To implement the ML supervised training of NN–HMM parameters $\mathbf{w}$, we first perform Viterbi decoding of training speech $\mathbf{O} = \{\mathbf{o}_t | t = 1, \cdots, T\}$ by using true transcription and NN–HMM parameters Θ or $\mathbf{w}$ at the current iteration. Each frame $\mathbf{o}_t$ is assigned an associated HMM state or output neuron $s_t = k$. This Viterbi alignment is equivalent to assigning a label or target value $t_{tk} \triangleq \gamma_t(k)$ of a speech frame $\mathbf{o}_t$ at an output neuron k of either 0 or 1. That is, $\gamma_t(k) \in \{0, 1\}$ is treated as a Bernoulli target variable. The negative auxiliary function in ML estimation is accordingly viewed as the *cross entropy error function* between the Bernoulli target values based on current estimate $\mathbf{w}$ and the posterior distributions from the NN outputs using new estimate $\mathbf{w}'$. By minimizing the cross entropy between the desired outputs $\{\gamma_t(k)\}$ and the actual outputs $\{y_k(\mathbf{o}_t, \mathbf{w})\}$ over all time frames $1 \leq t \leq T$ and all context-dependent HMM states $1 \leq k \leq K$, we establish the discriminative acoustic models (NN–HMMs) for speech recognition. The ML NN–HMM parameters $\Theta^{\mathrm{ML}} = \{\mathbf{w}^{\mathrm{ML}}\}$ are estimated. The error back-propagation algorithm can be implemented to train the synaptic weights $\mathbf{w}^{\mathrm{ML}}$ by minimizing $-Q(\mathbf{w}'|\mathbf{w})$, as given in Eq. (6.66).

More importantly, we address the maximum a-posteriori (MAP) estimation of NN–HMM parameters where the uncertainty of synaptic weights is compensated by using a prior distribution. For simplicity, we assume that the continuous values of weights in different layers come from a multivariate Gaussian distribution,

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w), \tag{6.67}$$

where $\boldsymbol{\mu}_w$ denotes the mean vector and $\boldsymbol{\Sigma}_w$ denotes the covariance matrix. This prior information could be empirically inferred from training data. There are two Bayesian inference stages in the so-called Bayesian NN–HMMs. The first stage is to find the *point estimate* of synaptic weights according to the MAP estimation. This idea is similar to MAP estimation of HMM parameters (Gauvain & Lee 1994). MAP estimates of NN–HMM parameters are calculated for adaptive training or speaker adaptation. The second stage is to conduct the *distribution estimation*, and this stage tries to fulfil a full Bayesian analysis by calculating the marginal likelihood with respect to all values of synaptic weights. Robustness of speech recognition to variations of synaptic weights due to heterogeneous data is assured.

In the first inference stage, MAP estimates of model parameters $\Theta^{\mathrm{MAP}} = \{\mathbf{w}^{\mathrm{MAP}}\}$ are calculated by maximizing a-posteriori probability or the product of likelihood function and prior distribution. An EM algorithm is applied to find MAP estimates by maximizing the posterior auxiliary function $Q^{\mathrm{MAP}}(\Theta'|\Theta)$. A MAP estimate of synaptic weights $\mathbf{w}^{\mathrm{MAP}}$ is derived by minimizing the corresponding auxiliary function:

$$\begin{aligned} -Q^{\mathrm{MAP}}(\mathbf{w}'|\mathbf{w}) \propto & -\sum_{t=1}^{T}\sum_{k=1}^{K} \gamma_t(k) \log p(s_t = k|\mathbf{o}_t, \mathbf{w}) \\ & + \frac{1}{2}(\mathbf{w}' - \boldsymbol{\mu}_w)^{\intercal} \boldsymbol{\Sigma}_w^{-1} (\mathbf{w}' - \boldsymbol{\mu}_w), \end{aligned} \tag{6.68}$$

which is viewed as a kind of penalized cross entropy error function. The prior distribution provides subjective information for Bayesian learning, or equivalently, serves as a

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

penalty function for model training. $\boldsymbol{\Sigma}_w$ plays the role of a metric matrix of the $\mathbf{w}' - \boldsymbol{\mu}_w$ distance. Again, the error back-propagation algorithm is applied to estimate the synaptic weights $\mathbf{w}'$ at a new iteration. Using this algorithm, the derivative of penalty function with respect to $\mathbf{w}'$ contributes the estimation of $\mathbf{w}^{\text{MAP}}$. The estimated $\mathbf{w}^{\text{MAP}}$ is treated as a point estimate or deterministic value for prediction or classification of future data.

6.4.3 Laplace approximation for Bayesian neural networks

In the second Bayesian inference stage, the synaptic weights $\mathbf{w}$ are treated as a latent random variable. The *hyperparameters* $\Psi = \{\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w\}$ are used to represent the uncertainty of random parameters $\mathbf{w}$. Such uncertainty information plays an important role in subjective inference, adaptive learning, and robust classification. Similarly to the BPC based on the standard HMMs as addressed in Section 6.3, we would like to present the BPC decision rule based on the combined NN–HMMs. To do so, we need to calculate the predictive distribution of test utterance $\mathbf{O}$ which is marginalized by considering all values of weight parameters $\mathbf{w}$ in the likelihood function $p(\mathbf{O}|\mathbf{w})$:

$$\begin{aligned}\tilde{p}(\mathbf{O}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) &= \int p(\mathbf{O}, \mathbf{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)d\mathbf{w} \\ &= \int p(\mathbf{O}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)d\mathbf{w} \\ &\triangleq \mathbb{E}_{(\mathbf{w})}[p(\mathbf{O}|\mathbf{w})]. \end{aligned} \tag{6.69}$$

The integral is performed over the parameter space of $\mathbf{w}$. However, due to the non-linear feed-forward network function $y_k(\mathbf{o}_t, \mathbf{w}) = p(s_t = k|\mathbf{o}_t, \mathbf{w})$, the closed-form solution to the integral calculation does not exist and should be carried out by using the Laplace approximation. By applying the Laplace approximation for integral operation in BIC and BPC decision, as given in Eqs. (6.16) and (6.43), we develop the BPC decision based on the NN–HMMs according to the approximate predictive distribution:

$$\begin{aligned}\tilde{p}(\mathbf{O}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) &\approx p(\mathbf{O}|\mathbf{w}^{\text{MAP}}) \cdot p(\mathbf{w}^{\text{MAP}}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) \cdot |\mathbf{H}|^{-1/2} \\ &= p(\mathbf{w}^{\text{MAP}}|\mathbf{O}, \boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) \cdot |\mathbf{H}|^{-1/2} \\ &\approx \exp\left(Q^{\text{MAP}}(\mathbf{w}'|\mathbf{w})\right)|_{\mathbf{w}'=\mathbf{w}^{\text{MAP}}} \cdot |\mathbf{H}|^{-1/2}, \end{aligned} \tag{6.70}$$

where the mode $\mathbf{w}^{\text{MAP}}$ of posterior distribution $p(\mathbf{w}|\mathbf{O}, \boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$ is obtained through the EM iteration procedure,

$$\mathbf{w}^{\text{MAP}} = \arg\max_{\mathbf{w}'} \{Q^{\text{MAP}}(\mathbf{w}'|\mathbf{w})\}, \tag{6.71}$$

and the Hessian matrix $\mathbf{H}$ is also calculated according to the EM algorithm through

$$\begin{aligned}\mathbf{H} &= \nabla_{\mathbf{w}'}\nabla_{\mathbf{w}'} \log\{p(\mathbf{O}|\mathbf{w}')p(\mathbf{w}'|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)\}|_{\mathbf{w}'=\mathbf{w}^{\text{MAP}}} \\ &\approx \nabla_{\mathbf{w}'}\nabla_{\mathbf{w}'} Q^{\text{MAP}}(\mathbf{w}'|\mathbf{w})|_{\mathbf{w}'=\mathbf{w}^{\text{MAP}}} \\ &= \nabla_{\mathbf{w}'}\nabla_{\mathbf{w}'} Q^{\text{ML}}(\mathbf{w}'|\mathbf{w})|_{\mathbf{w}'=\mathbf{w}^{\text{MAP}}} - \boldsymbol{\Sigma}_w^{-1}, \end{aligned} \tag{6.72}$$

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

which is the second-order differentiation of the log posterior distribution with respect to NN–HMM parameters at the mode $\mathbf{w}^{\mathrm{MAP}}$. Here, the auxiliary function $Q^{\mathrm{MAP}}(\mathbf{w}'|\mathbf{w})$ is introduced because the NN–HMM framework involves the missing data problem. As explained in Section 6.3.3, the EM algorithm is applied to iteratively approximate the posterior distribution at its mode by

$$p(\mathbf{w}'|\mathbf{O}, \boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)|_{\mathbf{w}'=\mathbf{w}^{\mathrm{MAP}}} \approx \exp\left(Q^{\mathrm{MAP}}(\mathbf{w}'|\mathbf{w})\right)|_{\mathbf{w}'=\mathbf{w}^{\mathrm{MAP}}}. \tag{6.73}$$

Accordingly, we construct the BPC decision based on NN–HMMs, which can achieve robustness in automatic speech recognition.

6.5 Decision tree clustering

This section introduces the application of BIC-based model selection to decision tree clustering of context-dependent HMMs, as performed in Shinoda & Watanabe (1997) and Chou & Reichl (1999), without having to set a heuristic stopping criterion, as is done in Young, Odell & Woodland (1994).[4] This section first describes decision tree clustering based on the ML criterion, which determines the tying structure of context-dependent HMMs efficiently. Then, the approach is extended by using BIC.

6.5.1 Decision tree clustering using ML criterion

The decision tree method has been widely used to construct a tied state HMM effectively by utilizing the phonetic knowledge-based constraint and the binary tree search (Odell 1995) approaches. Here, we introduce a conventional decision tree method using the ML criterion.

Let $\Omega(n)$ denote a set of states that tree node n holds. We start with only a root node ($n = 0$) which holds a set of all context-dependent phone (e.g., triphone hereinafter, as shown in Figure 6.5) HMM states $\Omega(0)$ for an identical center phoneme. The set of triphone states is then split into two sets depending on question Q, $\Omega(n_Y^Q)$ and $\Omega(n_N^Q)$, which are held by two new nodes, n_Y^Q and n_N^Q, respectively, as shown in Figure 6.3. The partition is determined by an answer to a phonemic question Q, such as "is the preceding phoneme a vowel" and "is the following phoneme a nasal." A particular question is chosen so that the partition is the optimal of all the possibilities, based on the likelihood value. We continue this splitting successively for every new set of states to obtain a binary tree, as shown in Figure 6.4, where each leaf node holds a shared set of triphone states. The states belonging to the same cluster are merged into a single HMM state. A set of triphones is thus represented by a set of tied-state HMMs. An HMM in an acoustic model usually has a left-to-right topology with three or four temporal states. A decision tree is usually produced for each state in the sequence, and

[4] Acoustic model selections based on BIC and minimum description length (MDL) criteria have been independently proposed, but they are practically the same if we deal with Gaussian distributions. Therefore, they are identified in this book and referred to as BIC.

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

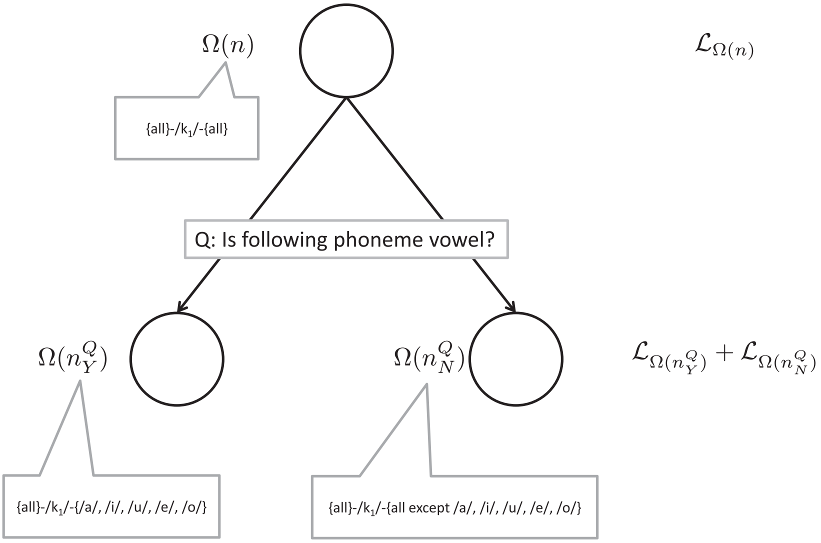
$\Omega(n)$
{all}-/k$_1$/-{all}
$\mathcal{L}_{\Omega(n)}$
Q: Is following phoneme vowel?
$\Omega(n_Y^Q)$
$\Omega(n_N^Q)$
$\mathcal{L}_{\Omega(n_Y^Q)} + \mathcal{L}_{\Omega(n_N^Q)}$
{all}-/k$_1$/-{/a/, /i/, /u/, /e/, /o/}
{all}-/k$_1$/-{all except /a/, /i/, /u/, /e/, /o/}

Figure 6.3 Splitting a set of HMM states $\Omega(n)$ in node n into two sets of states $\Omega(n_Y^Q)$ and $\Omega(n_N^Q)$ according to question Q. The objective function is changed from $\mathcal{L}_{\Omega(n)}$ to $\mathcal{L}_{\Omega(n_Y^Q)} + \mathcal{L}_{\Omega(n_N^Q)}$ after the split.

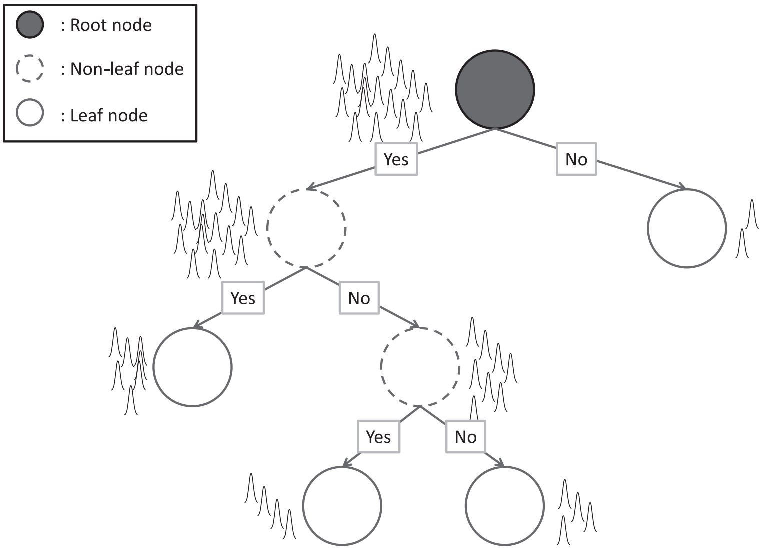

Figure 6.4 Binary decision tree. A context-dependent HMM state is represented by a single Gaussian distribution, and a set of states is assigned to each node. Two child nodes are obtained based on a yes/no answer to a (phonemic) question.

the trees are independent of each other, or a single decision tree is produced for all states.

The phonetic question concerns the preceding and following phoneme context, and is obtained through knowledge of the phonetics (Odell 1995) or it is also decided

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

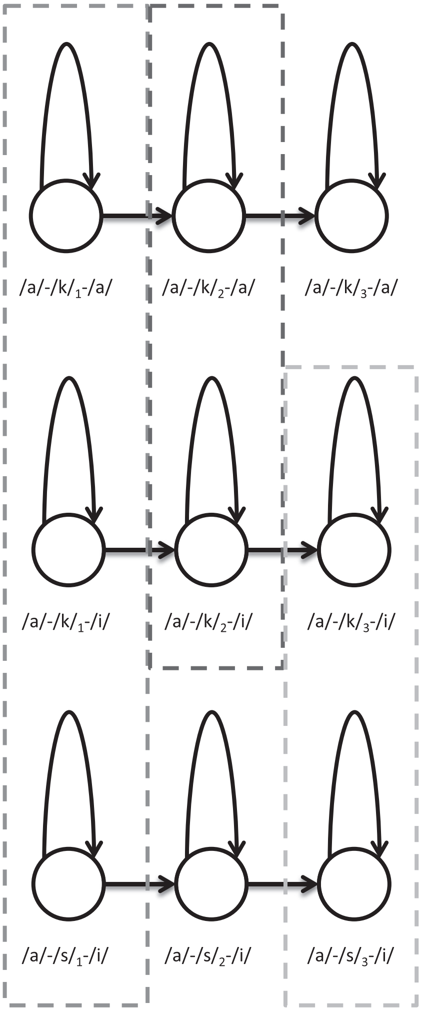

Figure 6.5 Examples of context-dependent (triphone) HMM and sharing structure. We prepare a three-state left-to-right HMM for each triphone, where /a/-/k/$_j$-/a/ denotes a jth state in a triphone HMM with preceding phoneme /a/, central phoneme /k/, and following phoneme /a/. In this example, triphone HMM states /a/-/k/$_1$-/a/, /a/-/k/$_1$-/i/, and /a/-/s/$_1$-/i/ are shared with the same Gaussian distribution.

in a data-driven manner (Povey, Ghoshal, Boulianne *et al.* 2011). Table 6.1 shows examples of the question. In a conventional ML-based approach, an appropriate question is obtained by maximizing a likelihood value as follows:

$$Q = \arg\max_{Q'} \Delta\mathcal{L}_{(Q')}, \tag{6.74}$$

where $\Delta\mathcal{L}_{(Q)}$ denotes the gain of log-likelihood when a state set in a node is split by a question Q. To calculate $\Delta\mathcal{L}_{(Q)}$, we assume the following constraints:

Table 6.1 Examples of phonetic questions for a $j(=1)$th HMM state of a phoneme /a/.

Question	Yes	No
Preceding phoneme is vowel?	{/a/, /i/, /u/, /e/, /o/} - $/\text{a}/_{j=1}$ - { all }	otherwise
Following phoneme is vowel?	{ all } - $/\text{a}/_{j=1}$ - {/a/, /i/, /u/, /e/, /o/}	otherwise
Following phoneme is media ?	{ all } - $/\text{a}/_{j=1}$ - {/b/, /d/, /g/}	otherwise
Preceding phoneme is back vowel?	{/u/, /o/} - $/\text{a}/_{j=1}$ - { all }	otherwise
⋮	⋮	⋮

- Data alignments $\gamma_t(j,k)$ and $\xi_t(i,j)$ for each state are fixed while splitting.
- Emission probability distribution in a state is represented by a single Gaussian distribution (i.e., $K = 1$).
- Covariance matrices have only diagonal elements.
- A contribution of state transitions a_{ij} and initial weights π_j for likelihood is disregarded.

These constraints simplify the likelihood calculation without using an iterative calculation, which greatly reduces the computational time.

Equation (6.74) corresponds to an approximation of the Bayes factor, as discussed in Eq. (6.12) to compare two models: M_1 models a set of triphones in the node n with a single Gaussian; and M_2 models two sets of triphones in the nodes n_Y^Q and n_N^Q separated by a question Q:

$$\begin{aligned}\Delta\mathcal{L}_{(Q')} &= \log\left(\frac{p(M_2|\mathbf{O})}{p(M_1|\mathbf{O})}\right)\\ &\approx \log\left(\frac{p(\mathbf{O}|\Theta^{\text{ML}}, M_2)}{p(\mathbf{O}|\Theta^{\text{ML}}, M_1)}\right).\end{aligned}\tag{6.75}$$

We obtain the gain of log-likelihood $\Delta\mathcal{L}_{(Q)}$ in Eq. (6.74) under the above constraints. Let $\mathbf{O}(i) = \{\mathbf{o}_t(i) \in \mathbb{R}^D : t = 1, \ldots, T(i)\}$ be a set of feature vectors that are assigned to HMM state i by the Viterbi algorithm. $T(i)$ denotes the frame number of training data assigned to state i, and D denotes the number of feature vector dimensions. From the constraints, log-likelihood $\mathcal{L}_\Omega$ for a training data set, assigned to state set Ω, is expressed by the following D dimensional Gaussian distribution:

$$\begin{aligned}\mathcal{L}_\Omega &= \log p(\{\mathbf{O}(i)\}_{i\in\Omega}|\boldsymbol{\mu}_\Omega, \boldsymbol{\Sigma}_\Omega)\\ &= \log\prod_{i\in\Omega}\prod_{t=1}^{T(i)}\mathcal{N}(\mathbf{O}_t(i)|\boldsymbol{\mu}_\Omega, \boldsymbol{\Sigma}_\Omega)\\ &= \log\prod_{i\in\Omega}\prod_{t=1}^{T(i)}(2\pi)^{-\frac{D}{2}}\,|\boldsymbol{\Sigma}_\Omega|^{-\frac{1}{2}}\,e^{-\frac{1}{2}(\mathbf{o}_t(i)-\boldsymbol{\mu}_\Omega)^\intercal\boldsymbol{\Sigma}_\Omega^{-1}(\mathbf{o}_t(i)-\boldsymbol{\mu}_\Omega)},\end{aligned}\tag{6.76}$$

where $\boldsymbol{\mu}_\Omega$ and $\boldsymbol{\Sigma}_\Omega$ denote a D dimensional mean vector and a $D \times D$ diagonal covariance matrix for a data set in Ω, respectively. ML estimates $\boldsymbol{\mu}_\Omega^{\mathrm{ML}}$ and $\boldsymbol{\Sigma}_\Omega^{\mathrm{ML}}$ are obtained by using the derivative technique. First, Eq. (6.76) is rewritten as

$$\mathcal{L}_\Omega = -\frac{D}{2}\log(2\pi)\sum_{i\in\Omega} T(i) - \frac{1}{2}\sum_{i\in\Omega}\sum_{t=1}^{T(i)}\sum_{d=1}^{D}\left(\log \Sigma_{\Omega d} + \frac{(o_{td}(i) - \mu_{\Omega d})^2}{\Sigma_{\Omega d}}\right), \tag{6.77}$$

where $\Sigma_{\Omega d}$ is a d-d element of the diagonal covariance matrix $\boldsymbol{\Sigma}_\Omega$. Then, by using the following derivative with respect to $\mu_{\Omega d}$,

$$\frac{\partial}{\partial \mu_{\Omega d}}\mathcal{L}_\Omega = -\frac{1}{2}\sum_{i\in\Omega}\sum_{t=1}^{T(i)} 2\frac{o_{td}(i) - \mu_{\Omega d}}{\Sigma_{\Omega d}} = 0, \tag{6.78}$$

we can obtain the ML estimate of the mean vector $\boldsymbol{\mu}_\Omega^{\mathrm{ML}}$ as:

$$\boldsymbol{\mu}_\Omega^{\mathrm{ML}} = \frac{\sum_{i\in\Omega}\sum_{t=1}^{T(i)} \mathbf{o}_t(i)}{T_\Omega}. \tag{6.79}$$

Similarly, by using the following derivative with respect to $\Sigma_{\Omega d}$,

$$\frac{\partial}{\partial \Sigma_{\Omega d}}\mathcal{L}_\Omega = -\frac{1}{2}\sum_{i\in\Omega}\sum_{t=1}^{T(i)}\left(\frac{1}{\Sigma_{\Omega d}} - \frac{(o_{td}(i) - \mu_{\Omega d})^2}{\Sigma_{\Omega d}^2}\right) = 0, \tag{6.80}$$

we can obtain the ML estimate of the diagonal component of the covariance matrix $\Sigma_{\Omega d}^{\mathrm{ML}}$ as:

$$\Sigma_{\Omega d}^{\mathrm{ML}} = \frac{\sum_{i\in\Omega}\sum_{t=1}^{T(i)}(o_{td}(i) - \mu_{\Omega d}^{\mathrm{ML}})^2}{T_\Omega}. \tag{6.81}$$

We summarize the above ML estimates as:

$$\begin{aligned}
\boldsymbol{\mu}_\Omega^{\mathrm{ML}} &= \frac{\sum_{i\in\Omega}\sum_{t=1}^{T(i)} \mathbf{o}_t(i)}{T_\Omega}, \\
[\boldsymbol{\Sigma}_\Omega^{\mathrm{ML}}]_{dd} = \Sigma_{\Omega d}^{\mathrm{ML}} &= \frac{\sum_{i\in\Omega}\sum_{t=1}^{T(i)}(o_{td}(i) - \mu_{\Omega d}^{\mathrm{ML}})^2}{T_\Omega}.
\end{aligned} \tag{6.82}$$

$T_\Omega \triangleq \sum_{i\in\Omega} T(i)$ denotes the frame number of training data assigned to HMM states belonging to Ω. $\Sigma_{\Omega d}^{\mathrm{ML}}$ denotes a d-d component for matrix $\boldsymbol{\Sigma}_\Omega^{\mathrm{ML}}$. By substituting Eq. (6.82) into Eq. (6.76), we can derive the following log-likelihood $\mathcal{L}_\Omega$ with the ML estimates of $\boldsymbol{\Sigma}_\Omega^{\mathrm{ML}}$:

$$\begin{aligned}
\mathcal{L}_\Omega &= -\frac{D}{2}\log(2\pi)\sum_{i\in\Omega} T(i) - \frac{1}{2}\sum_{i\in\Omega}\sum_{t=1}^{T(i)}\sum_{d=1}^{D}\left(\log \Sigma_{\Omega d} + \frac{(o_{td}(i) - \mu_{\Omega d})^2}{\Sigma_{\Omega d}}\right)\Bigg|_{\substack{\mu_{\Omega d}\to\mu_{\Omega d}^{\mathrm{ML}} \\ \Sigma_{\Omega d}\to\Sigma_{\Omega d}^{\mathrm{ML}}}} \\
&= -\frac{D}{2}\log(2\pi)T_\Omega - \frac{1}{2}T_\Omega\sum_{d=1}^{D}\log \Sigma_{\Omega d}^{\mathrm{ML}} - \frac{1}{2}T_\Omega\sum_{d=1}^{D}\frac{\Sigma_{\Omega d}^{\mathrm{ML}}}{\Sigma_{\Omega d}^{\mathrm{ML}}} \\
&= -\frac{T_\Omega}{2}\left(D(1 + \log(2\pi)) + \log\left|\boldsymbol{\Sigma}_\Omega^{\mathrm{ML}}\right|\right).
\end{aligned} \tag{6.83}$$

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

Note that the likelihood value is represented by the determinant of the ML estimate of the covariance matrix, which can easily be calculated as we use the diagonal covariance.

Therefore, a gain of log-likelihood $\Delta\mathcal{L}_{(Q)}$ can be solved as follows (Odell 1995):

$$\begin{aligned}\Delta\mathcal{L}_{(Q)} &= \mathcal{L}_{\Omega(n_Y^Q)} + \mathcal{L}_{\Omega(n_N^Q)} - \mathcal{L}_{\Omega(n)} \\ &= l(\Omega(n_Y^Q)) + l(\Omega(n_N^Q)) - l(\Omega(n)).\end{aligned} \tag{6.84}$$

Here l in Eq. (6.84) is defined as:

$$\begin{aligned}l(\Omega) &\triangleq -\frac{1}{2}\left(T_\Omega \log\left(\left|\mathbf{\Sigma}_\Omega^{\mathrm{ML}}\right|\right)\right) \\ &\text{for } \Omega = \{\Omega(n_Y^Q), \Omega(n_N^Q), \Omega(n)\},\end{aligned}$$

where we use the following relation:

$$T_{\Omega(n)} = T_{\Omega(n_Y^Q)} + T_{\Omega(n_N^Q)}. \tag{6.85}$$

Equations (6.84) and (6.85) show that $\Delta\mathcal{L}_{(Q)}$ can be calculated using the ML estimate $\mathbf{\Sigma}_\Omega^{\mathrm{ML}}$ and frame number T_Ω. Finally, the appropriate question in the sense of the ML criterion can be computed by substituting Eq. (6.84) into Eq. (6.74).

However, we cannot use this likelihood criterion for stopping the split of nodes in a tree. That is, the positivity of $\Delta\mathcal{L}_{(Q)}$ for any split causes the ML criterion to always select the model structure in which the number of states is the largest. That is, no states are shared at all. To avoid this, the ML criterion requires the following threshold to be set to stop splitting manually:

$$\Delta\mathcal{L} \leq \mathrm{Thd}. \tag{6.86}$$

There exist other approaches to stopping splitting manually by setting the number of total states, or the maximum depth of the tree, as well as a hybrid approach combining those approaches. However, the effectiveness of the thresholds in all of these manual approaches has to be judged by the performance of the development set.

6.5.2 Decision tree clustering using BIC

We consider automatic model selection based on the BIC criterion, which is widely used in model selection for various aspects of statistical modeling. Recall the following definition of a BIC-based objective function for $\mathbf{O} = \{\mathbf{o}_t | t = 1, \cdots T\}$ in Eq. (6.29):

$$\mathcal{L}^{\mathrm{BIC}} = \log p(\mathbf{O}|\boldsymbol{\theta}^*, M) - \lambda\frac{J}{2}\log T, \tag{6.87}$$

where $\boldsymbol{\theta}^*$ is the ML or MAP estimate of model parameters for model M, J is the number of model parameters, and λ is a tuning parameter. Similarly to Section 6.5.1, we use a single Gaussian for each node. Then, Eq. (6.87) for node Ω can be rewritten as follows:

$$\begin{aligned}\mathcal{L}_\Omega^{\mathrm{BIC}} &= \mathcal{L}_\Omega - \lambda D \log T_\Omega \\ &= -\frac{T_\Omega}{2}\left(D(1+\log(2\pi)) + \log\left|\mathbf{\Sigma}_\Omega^{\mathrm{ML}}\right|\right) - \lambda D \log T_\Omega,\end{aligned} \tag{6.88}$$

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

where $J = 2D$ when we use the diagonal covariance matrix. Therefore, the gain of objective function $\Delta\mathcal{L}^{BIC}_{(Q)}$ using the BIC criterion (the logarithmic Bayes factor) is obtained while splitting a state set by question Q, as follows:

$$\Delta\mathcal{L}^{\mathrm{BIC}}_{(Q)} = \Delta\mathcal{L}_{(Q)} - \lambda D \log T_{\Omega(0)}, \tag{6.89}$$

where λ is a tuning parameter in the BIC criterion, and $T_{\Omega(0)}$ denotes the frame number of data assigned to a root node.[5] Equation (6.89) suggests that the BIC objective function penalizes the gain in log-likelihood on the basis of the balance between the number of free parameters and the amount of training data, and the penalty can be controlled by varying λ. Model structure selection is achieved according to the amount of training data by using $\Delta\mathcal{L}^{\mathrm{BIC}}_{(Q)}$ instead of using $\Delta\mathcal{L}_{(Q)}$ in Eq. (6.74), that is

$$Q = \arg\max_{Q'} \Delta\mathcal{L}^{\mathrm{BIC}}_{(Q')}. \tag{6.91}$$

We can also use $\Delta\mathcal{L}^{\mathrm{BIC}}_{(Q)}$ for stopping splitting when this value is negative, without using a threshold in Eq. (6.86), that is

$$\Delta\mathcal{L}^{\mathrm{BIC}}_{(Q)} \leq 0. \tag{6.92}$$

Thus, we can obtain the tied-state HMM by using the BIC model selection criterion.[6]

Note that the BIC criterion is an asymptotic criterion that is theoretically effective only when the amount of training data is sufficiently large. Therefore, in the case of a small amount of training data, model selection does not perform well because of the uncertain ML estimates $\boldsymbol{\mu}^{\mathrm{ML}}$ and $\boldsymbol{\Sigma}^{\mathrm{ML}}$. Shinoda & Watanabe (2000) show the effectiveness of the BIC criterion for decision tree clustering in a 5000 Japanese word recognition task by comparing the performance of acoustic models based on BIC with models based on heuristic stopping criteria (namely, the state occupancy count and the likelihood threshold). BIC selected 2069 triphone HMM states automatically with an 80.4 % recognition rate, while heuristic stopping criteria selected 1248 and 591 states with recognition rates of 77.9 % and 66.6 % in the best and worst cases, respectively. This result clearly shows the effectiveness of model selection using BIC. An extension of the BIC objective function by considering a tree structure is also discussed in Hu & Zhao (2007), and an extension based on variational Bayes is discussed in Section 7.3.6. In addition, BIC is used for Gaussian pruning in acoustic models Shinoda & Iso (2001), and speaker segmentation (Chen & Gopinath 1999), which is discussed in the next section.

[5] The following objective function

$$\Delta\mathcal{L}^{\mathrm{BIC}}_{(Q)} = \Delta\mathcal{L}_{(Q)} - \lambda D \log T_{\Omega(0)} \tag{6.90}$$

is derived in Shinoda & Watanabe (2000). There are several ways to set the penalty term from Eq. (6.89).

[6] The BIC criterion also has a tuning parameter λ in Eq. (6.87). However, the λ setting is more robust than the threshold setting in Eq. (6.86) (Shinoda & Watanabe 2000).

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

6.6 Speaker clustering/segmentation

BIC-based speaker segmentation is a particularly important technique for speaker diarization, which has been widely studied recently (Chen & Gopinath 1999, Anguera Miro, Bozonnet, Evans *et al.* 2012). The approach is usually performed in two steps; the first step segments a long sequence of speech to several chunks by using a model selection approach. Then the second step clusters these chunks to form speaker segments, which can also be performed by a model selection approach. Both steps use BIC as model selection. This section considers the first step of speaker segmentation based on the BIC criterion.

6.6.1 Speaker segmentation

First, we consider the simple segmentation problems of segmenting a speech sequence to two chunks where these chunks are uttered by two speakers, respectively, as shown in Figure 6.6. That is:

$$\{\mathbf{o}_1, \cdots, \mathbf{o}_T\} \rightarrow \{\mathbf{o}_1, \cdots, \mathbf{o}_t\} \text{ and } \{\mathbf{o}_{t+1}, \cdots, \mathbf{o}_T\}, \tag{6.93}$$

where t is a change point from one speaker to the other speakers or noises. Assuming that each speaker utterance is modeled by a single Gaussian, which is a very simple assumption but works well in a practical use, this problem can be regarded as a model selection problem where we have a set of candidates represented by:

- $M_0 : \mathbf{o}_1, \cdots, \mathbf{o}_T \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$;
- $M_t : \mathbf{o}_1, \cdots, \mathbf{o}_t \sim \mathcal{N}(\boldsymbol{\mu}_{1,t}, \boldsymbol{\Sigma}_{1,t}), \mathbf{o}_{t+1}, \cdots, \mathbf{o}_T \sim \mathcal{N}(\boldsymbol{\mu}_{2,t}, \boldsymbol{\Sigma}_{2,t})$
  for $t = [2, T-1]$.

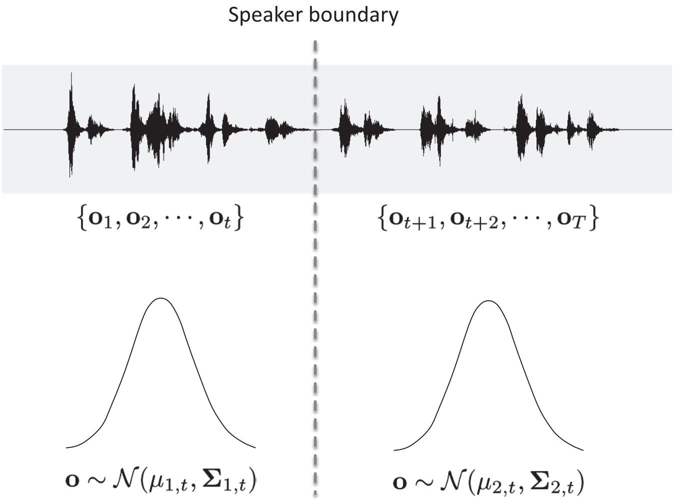
Speaker boundary
$\{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t\}$
$\{\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \cdots, \mathbf{o}_T\}$
$\mathbf{o} \sim \mathcal{N}(\mu_{1,t}, \mathbf{\Sigma}_{1,t})$
$\mathbf{o} \sim \mathcal{N}(\mu_{2,t}, \mathbf{\Sigma}_{2,t})$

Figure 6.6 Speaker segmentation for audio data, which find a speaker boundary t given speech features $\{\mathbf{o}_1, \cdots, \mathbf{o}_T\}$. The BIC segmentation method assumes that each segmentation is represented by a Gaussian distribution.

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

The model M_0 means that the sequence can be represented by one Gaussian with mean vector $\boldsymbol{\mu}$ and the diagonal covariance matrix $\boldsymbol{\Sigma}$, that is, the sequence only has one speaker utterance. Similarly, the model M_t for $t = [2, T-1]$ means that the sequence has two speakers with a change point t with two Gaussians that have mean vectors $\boldsymbol{\mu}_{1,t}$ and $\boldsymbol{\mu}_{2,t}$, and diagonal covariance matrices $\boldsymbol{\Sigma}_{1,t}$ and $\boldsymbol{\Sigma}_{2,t}$, depending on the change point t.

Based on these hypothesized models, we can also consider the Bayes factor in Eq. (6.12) to compare two models of M_0 and M_t:

$$\log\left(\frac{p(M_t|\mathbf{O})}{p(M_0|\mathbf{O})}\right) \text{ for } t = [2, T-1]. \tag{6.94}$$

Similarly to the decision tree clustering, we can select an appropriate model from M_0 and $\{M_t\}_{t=2}^{T-1}$ by using BIC. Recall the following definition of a BIC-based objective function of model M for $\mathbf{O} = \{\mathbf{o}_t | t = 1, \cdots N\}$ in Eq. (6.29):

$$\mathcal{L}^{\text{BIC}} = \log p(\mathbf{O}|\boldsymbol{\theta}^*, M) - \lambda\frac{J}{2}\log N, \tag{6.95}$$

where $\boldsymbol{\theta}^*$ is the ML or MAP estimate of model parameters for model M, J is the number of model parameters, and λ is a tuning parameter. Since we use a Gaussian (with diagonal covariance matrix) for the likelihood function in Eq. (6.95), the BIC function for M_0 is written as follows:

$$\begin{aligned}\mathcal{L}^{\text{BIC}}(M_0) &= \log\prod_{t=1}^{T}\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}^{\text{ML}}, \boldsymbol{\Sigma}^{\text{ML}}) - \lambda\frac{2D}{2}\log T \\ &= -\frac{T}{2}\left(D(1+\log(2\pi)) + \log\left|\boldsymbol{\Sigma}^{\text{ML}}\right|\right) - \lambda D\log T,\end{aligned} \tag{6.96}$$

where $\boldsymbol{\Sigma}^{\text{ML}}$ is the ML estimate of the diagonal covariance matrix:

$$\begin{aligned}\boldsymbol{\mu}^{\text{ML}} &= \frac{\sum_{t=1}^{T}\mathbf{o}_t}{T}, \\ \boldsymbol{\Sigma}_d^{\text{ML}} &= \frac{\sum_{t=1}^{T}(o_{td} - \mu_d^{\text{ML}})^2}{T}.\end{aligned} \tag{6.97}$$

This result is obtained by assuming that we only have one state in Eq. (6.83).

Similarly, M_t is written as follows:

$$\begin{aligned}\mathcal{L}^{\text{BIC}}(M_t) &= \log\prod_{t'=1}^{t}\mathcal{N}(\mathbf{o}_{t'}|\boldsymbol{\mu}_{1,t}^{\text{ML}}, \boldsymbol{\Sigma}_{1,t}^{\text{ML}}) + \log\prod_{t'=t+1}^{T}\mathcal{N}(\mathbf{o}_{t'}|\boldsymbol{\mu}_{2,t}^{\text{ML}}, \boldsymbol{\Sigma}_{2,t}^{\text{ML}}) - \lambda\frac{4D}{2}\log T \\ &= -\frac{t}{2}\left(D(1+\log(2\pi)) + \log\left|\boldsymbol{\Sigma}_{1,t}^{\text{ML}}\right|\right) \\ &\quad - \frac{T-t}{2}\left(D(1+\log(2\pi)) + \log\left|\boldsymbol{\Sigma}_{2,t}^{\text{ML}}\right|\right) - 2\lambda D\log T \\ &= -\frac{t}{2}\log\left|\boldsymbol{\Sigma}_{1,t}^{\text{ML}}\right| - \frac{T-t}{2}\log\left|\boldsymbol{\Sigma}_{2,t}^{\text{ML}}\right| - \frac{T}{2}(D(1+\log(2\pi))) - 2\lambda D\log T,\end{aligned} \tag{6.98}$$

where $\boldsymbol{\Sigma}_{1,t}^{\mathrm{ML}}$ and $\boldsymbol{\Sigma}_{2,t}^{\mathrm{ML}}$ are the ML estimates of the diagonal covariance matrices, and are obtained as follows:

$$\begin{aligned}
\boldsymbol{\mu}_{1,t}^{\mathrm{ML}} &= \frac{\sum_{t'=1}^{t} \mathbf{o}_{t'}}{t}, \\
\boldsymbol{\Sigma}_{1,td}^{\mathrm{ML}} &= \frac{\sum_{t'=1}^{t} (o_{t'd} - \mu_{1,td}^{\mathrm{ML}})^2}{t}, \\
\boldsymbol{\mu}_{2,t}^{\mathrm{ML}} &= \frac{\sum_{t'=t+1}^{T} \mathbf{o}_{t'}}{t}, \\
\boldsymbol{\Sigma}_{2,td}^{\mathrm{ML}} &= \frac{\sum_{t'=t+1}^{T} (o_{t'd} - \mu_{2,td}^{\mathrm{ML}})^2}{t}.
\end{aligned} \tag{6.99}$$

Again, this result is obtained by assuming that we only have two separated states (M_0 and M_t) in Eq. (6.83).

Therefore, the difference of BIC values between M_0 and M_t is represented as follows:

$$\begin{aligned}
\Delta\mathcal{L}^{\mathrm{BIC}}(t) &\triangleq \mathcal{L}^{\mathrm{BIC}}(M_t) - \mathcal{L}^{\mathrm{BIC}}(M_0) \\
&= -\frac{T}{2}\log\left|\boldsymbol{\Sigma}^{\mathrm{ML}}\right| + \frac{t}{2}\log\left|\boldsymbol{\Sigma}_{1,t}^{\mathrm{ML}}\right| + \frac{T-t}{2}\log\left|\boldsymbol{\Sigma}_{2,t}^{\mathrm{ML}}\right| - \lambda D \log T.
\end{aligned} \tag{6.100}$$

This is the objective function of finding the segmentation boundary based on the BIC criterion. If the $\Delta\mathcal{L}^{\mathrm{BIC}}(t)$ is positive, t is a segmentation boundary, and the optimal boundary $\hat{t}$ is obtained by maximizing $\Delta\mathcal{L}^{\mathrm{BIC}}(t)$ as follows:

$$\hat{t} = \arg\max_{t} \Delta\mathcal{L}^{\mathrm{BIC}}(t) \text{ if } \Delta\mathcal{L}^{\mathrm{BIC}}(t) > 0. \tag{6.101}$$

Thus, we can find the optimal boundary of two speech segments by using BIC. If we consider multiple changing points based on BIC, we can use Algorithm 9. The approach has also been extended to deal with prior distributions (Watanabe & Nakamura 2009).

Algorithm 9 Detecting multiple changing points

```
Require: interval $[t_a, t_b] : t_a = 1; t_b = 2$
while $t_a < T$ do
    Detect if there is one changing point in $[t_a, t_b]$ via BIC
    if no change in $[t_a, t_b]$ then
        $t_b = t_b + 1$
    else
        $t_a = \hat{t} + 1; t_b = t_a + 1$
    end if
end while
```

6.6.2 Speaker clustering

Once we have the speech segments obtained by the BIC-based speech segmentation, we can again use the BIC criterion to cluster the speech segments, where the cluster obtained can be interpreted as a specific speaker. This can be obtained by similar techniques to the decision tree clustering of context-dependent HMMs in Section 6.5.2. The main difference between them is that the decision tree clustering is performed by *splitting* the nodes from the root node by using the (phonetic) question, while the speaker clustering starts to *merge* the leaf nodes, which are represented by a speech segment, successively to build a tree.

Let i be a speech segment index obtained by speech segmentation techniques, and we have in total J segments. Then, similarly to the previous sections, we can compute the approximated Bayes factor based on the following difference of the BIC value when merging the speech segments i and i' as

$$\Delta\mathcal{L}^{\mathrm{BIC}}_{i,i'} \triangleq \mathcal{L}^{\mathrm{BIC}}_{i,i'} - \mathcal{L}^{\mathrm{BIC}}_{i} + \mathcal{L}^{\mathrm{BIC}}_{i'}, \tag{6.102}$$

where the BIC values of $\mathcal{L}^{\mathrm{BIC}}_{i}$ and $\mathcal{L}^{\mathrm{BIC}}_{i'}$ are represented as:

$$\mathcal{L}^{\mathrm{BIC}}_{i} = -\frac{T_i}{2}\left(D(1+\log(2\pi)) + \log\left|\mathbf{\Sigma}^{\mathrm{ML}}_{i}\right|\right) - \lambda D \log T_i, \tag{6.103}$$

where T_i and $\mathbf{\Sigma}^{\mathrm{ML}}_{i}$ are the number of frames assigned to the speech segment i and the ML estimate of the covariance matrix of the speech segment i, respectively. Similarly, the BIC value of $\mathcal{L}^{\mathrm{BIC}}_{i,i'}$, which merges two nodes i and i', is represented as

$$\mathcal{L}^{\mathrm{BIC}}_{i,i'} = -\frac{T_{i,i'}}{2}\left(D(1+\log(2\pi)) + \log\left|\mathbf{\Sigma}^{\mathrm{ML}}_{i,i'}\right|\right) - \lambda D \log T_{i,i'}, \tag{6.104}$$

where $T_{i,i'} = T_i + T_{i'}$, and $\mathbf{\Sigma}^{\mathrm{ML}}_{i,i'}$ is the ML estimate of the covariance matrix when we represent the speech segments i and i' as a single Gaussian.

The most appropriate combination of merging two nodes can be obtained by considering all possible i and i' combinations and selecting the combination

$$\hat{i}, \hat{i}' = \arg \max_{1\leq i\leq J, i<i'<J} \Delta\mathcal{L}^{\mathrm{BIC}}_{i,i'}. \tag{6.105}$$

And the selected $\hat{i}$ and $\hat{i}'$ can be merged to a new node. This process is iteratively performed while it satisfies $\Delta\mathcal{L}^{\mathrm{BIC}}_{i,i'} > 0$. The final leaf nodes can represent a speaker cluster. This approach is also called *agglomerative clustering*, and BIC based agglomerative clustering is actually used to build state-of-the-art speaker diarization systems (Wooters & Huijbregts 2008), which provide the "who spoke when" information in speech data.

6.7 Summary

This chapter describes the asymptotic approximation of the Bayesian approach, and introduces the Laplace approximation and the BIC criterion. Both approaches are very powerful for approximating the Bayesian inference of acoustic model parameters in speech recognition and Bayesian inference of model selection problems in speech

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press

clustering and segmentation. However, the asymptotic approximation makes the single Gaussian assumption for the prior and posterior distributions and large sample assumptions for target data, which limits the application for speech and language processing. The next chapter provides another approximation technique, variational Bayes, which can handle the problem in the asymptotic approximation.

https://doi.org/10.1017/CBO9781107295360.007 Published online by Cambridge University Press