CS281B/Stat241B: Advanced Topics in Learning & Decision Making

Approximation of Marginal Likelihood

Lecturer: Michael I. Jordan *Scribes: Shawn Chang and Barry Chen*

Approximation methods may be used in situations where we are not able to calculate the exact marginal likelihoods. The methods for approximating marginal likelihoods can be divided into two categories: the within-model approximation and between-model approximation.

1 Within-Model Approximation

- BIC/Laplace/Cheeseman-Stutz (see paper by Chickering and Heckerman, and paper by DiCiccio et. al.)

- Chib's method:

$$p(\mathcal{D}|M) = \frac{p(\mathcal{D}|\theta_0, M)p(\theta_0|M)}{p(\theta_0|\mathcal{D}, M)} \tag{1}$$

  where we choose some θ_0 such as the MAP estimate $\hat{\theta}_{MAP}$. The likelihood term $p(\mathcal{D}|\theta_0, M)$ may be obtained by the Junction Tree algorithm, the prior term $p(\theta_0|M)$ by table lookup, and the denominator by simple MCMC approximation.

- Variational Methods (Variational Bayes): This is a non-sampling method. Let $q(\theta|M)$ be some proposal distribution simpler than $p(\theta|\mathcal{D}, M)$. We use the $q(\theta|M)$ to calculate an upper bound on the marginal likelihood.

$$\begin{aligned}
\log p(\mathcal{D}|M) &= \log \int p(\mathcal{D}, \theta|M) d\theta && (2)\\
&= \log \int q(\theta|M) \frac{p(\mathcal{D}, \theta|M)}{q(\theta|M)} d\theta && (3)\\
&\geq \int q(\theta|M) \log \frac{p(\mathcal{D}, \theta|M)}{q(\theta|M)} d\theta && (4)
\end{aligned}$$

  where the last step is by Jensen's inequality, and it is exact if $q(\theta|M) = p(\theta|\mathcal{D}, M)$.

- Sampling methods, e.g. Naive Sampling:

$$\begin{aligned}
p(\mathcal{D}|M) &= \int p(\mathcal{D}|\theta, M)p(\theta|M) d\theta && (5)\\
&\approx \frac{1}{T} \sum_{t=1}^{T} p(\mathcal{D}|\theta^{(t)}, M) && (6)\\
& && (7)
\end{aligned}$$

  where $\theta^{(t)} \overset{i.i.d.}{\sim} p(\theta|M)$. However, the sampling becomes difficult if the prior $p(\theta|M)$ is diffuse.

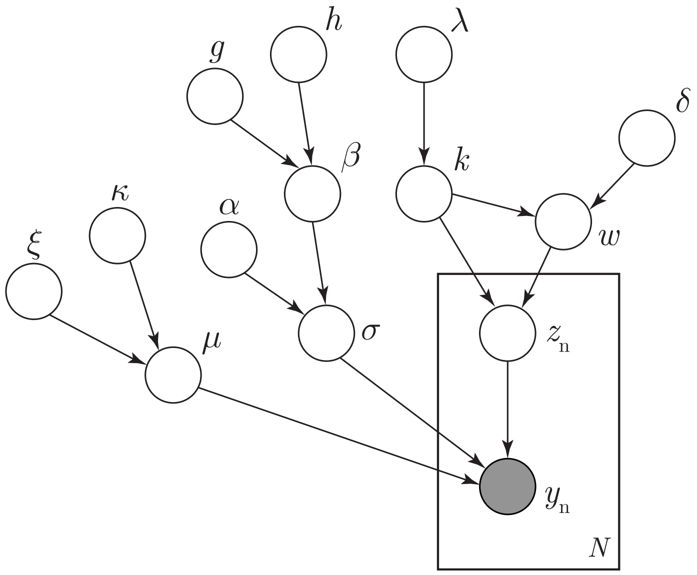

Figure 1: Graphical model representation of a Gaussian Mixture Model. The shaded node is observed, and the box surrounding z_n and y_n denotes that there are N replicates of them.

2 Between-Model Approximation

Reversible Jump (RJ) MCMC is a commonly used between-model approximation method. Suppose we are to choose k, the number of mixtures, in some mixture model. For different values of k, the model changes its number of parameters, so it becomes difficult to compute the marginal likelihoods within the same model. RJ MCMC and other between-model methods provide a way to sample among different models (corresponding to different k's) in a uniform fashion. Let's illustrate the idea by an example of a univariate Gaussian Mixture Model (GMM) with the following parameterization:

$$y_i \sim \sum_j w_j f(\cdot|\theta_j) \tag{8}$$

$$\theta_j = (\mu_j, \sigma_j^2) \tag{9}$$

$$w = \mathrm{Dir}(\delta, \delta, \ldots, \delta) \tag{10}$$

$$k \sim \mathrm{Poisson}(\lambda) \tag{11}$$

$$\sigma_j^{-2} \sim \Gamma(\alpha, \beta) \tag{12}$$

$$\mu_j \sim \mathrm{Gaussian}(\xi, \kappa) \tag{13}$$

$$\beta \sim \Gamma(g, h) \tag{14}$$

Figure 1 shows the graphical model representation of the GMM. The shaded node is observed, and the box surrounding z_n and y_n denotes that there are N replicates of them. The nodes δ, λ, κ, ξ,g, and h are fixed. However, experiments show that the analysis of the model is robust with different choices of those parameters (and this is the reason why we parameterize the model at this high level).

The priors on nodes are chosen carefully to be their conjugate prior distributions, i.e. the product of likelihood and conjugate prior stays within the same family as the likelihood. For example, the Gaussian likelihood $e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$ takes the Gaussian prior $e^{-\frac{1}{2}\kappa(\mu-\xi)^2}$, which is the conjugate prior for Gaussian itself. Interestingly, when we put mixtures on both the likelihood and priors, the conjugacy still holds.

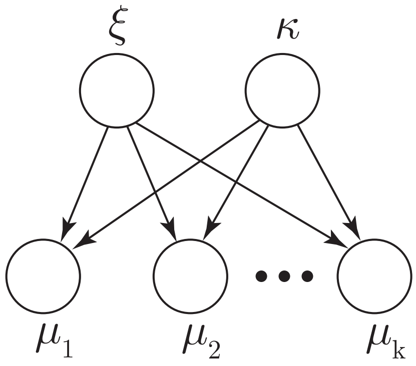

Figure 2: All μ_j's share the same distribution $N(\xi, \kappa)$. This results in a smoothing effect.

Note that since all μ_j's use same $N(\xi, \kappa)$ distribution (as shown in Figure 2), it effectively performs smoothing (or sharing of data by pulling μ_j's close to each other).