Markov Chain Monte Carlo

Diego Garrido

Markov Chain Monte Carlo (MCMC) is one of the most used methods to approximate a complicated distribution. In this paper we use two MCMC algorithm, the Metropolis-Hastings algorithm and the Gibbs Sampling algorithm.

Jupyter Notebook

1 Metropolis Hastings

In this section we use the Metropolis-Hastings (MH) algorithm to approximate empirically the distribution $p(x) \propto 2e^{-(x-2)^2} + e^{-|x|} + e^{-(x+2)^2}e^{-|x+2|} = \tilde{p}(x)$. Firstly, we chose a proposal distribution with the same support as $p(x)$, as a Gaussian proposal, i.e., $q(x^{'}|x_i) = \mathcal{N}(x_i, \sigma^2)$. Since the proposal is symmetric, the acceptance probability is given by $\alpha = \min\left(1, \frac{\tilde{p}(x')}{\tilde{p}(x_i)}\right)$. Secondly, we draws 5000 samples from $p(x)$ using MH with $\sigma \in \{0.01, 0.1, 1, 100\}$. This is ilustrated in Figure 1. Different choices of the proposal standard deviation σ lead to very different results. If the proposal is too narrow, only one mode of $p(x)$ might be visited. On the other hand, if it is too wide, the rejection rate can be very high, resulting in high correlations. In sum, the success or failure of the algorithm not only depends on the choice of the proposal distribution, it also depends on its parameters, in this case the adecuate σ might be in $\{0.1, 1\}$.

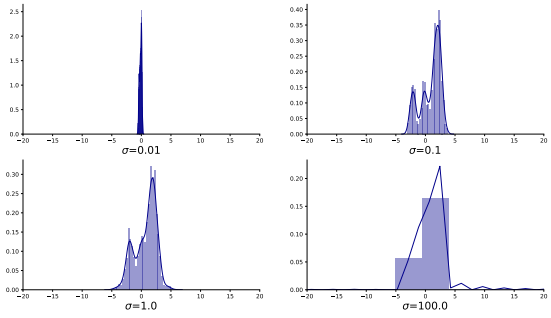

Figure 1: Approximations obtained using the MH algorithm with Gaussian proposal with different stanrdar deviation σ.

The output of the MH algotithm can be used for point estimation, estimating quantities as mean, median or mode (the point with highest density). However, that method is not much efficient to estimate the mode because the random samples solely rarely come from the vicinity of the mode. A better way to do that is Simulated Annealing (SA). This method sampling from $p^{1/T_i}(x)$ instead of $p(x)$. The term T_i is a decreasing cooling schedule with $\lim_{i\to\infty} T_i = 0$. To estimate the mode of $p(x)$ we set $T_i = Cln(i + T_0)$, with $T_0 = 1$ and $C = 2$, in this manner the series is decreasing. The result of applying SA to the previous example are shown in Figure 2. In this case the better approximation apparently is reached by $\sigma = 1$, where the mode is 1.95. In brief, to obtain efficient annealed algorithms, it is again important to choose suitable proposal distributions and an appropiate cooling schedule.

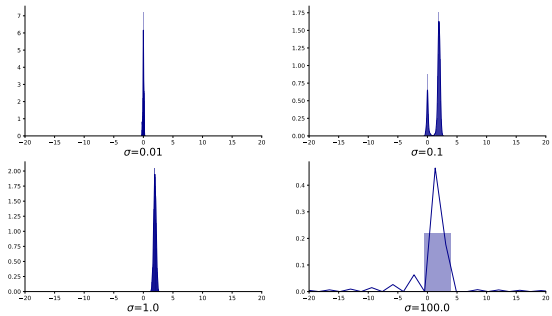

Figure 2: Discovering the modes of the target distribution using the SA algorithm.

2 Gibbs Sampling

In this section we use the Gibbs Sampling (GB) algorithm to empirically approximate $p(z, \pi, \mu|x)$, where $x = x_1, \ldots, x_{100}$ is the data, $z = z_1, \ldots, z_{100}$ are the hiddens variables, in this case are the binary cluster assignments, π is the mixture proportion, and $\mu = [\mu_0, \mu_1]$ are the parameters of the observation model. The generative process is detailed below:

$$\mu_0, \mu_1 \sim \mathcal{N}(0, 5) \tag{1}$$

$$\pi \sim Beta(0.8, 0.8) \tag{2}$$

$$z_1, \ldots, z_{100}|\mu, \pi \sim Bernoulli(\pi) \tag{3}$$

$$z_i \sim \mathcal{N}(\mu_{z_i,0.25}) \tag{4}$$

Firstly, we generate a random sample from this process. In this case the resulting main parameters are $\mu_0 =$

2.357, $mu_1 = -5.955$ and $\pi = 0.788$. This is ilustrated in Figure 3. Secondly, is necessary to get the full conditional distribution $p(z_i|z_{-i}, \pi, \mu, x)$, $p(\mu_k|z, \pi, \mu_{-k}, x)$ and $p(\pi|z, \mu, x)$. The variable z_i is conditional independent from z_{-i} and x_{-i} given x_i, so $p(z_i|z_{-i}, \pi, \mu, x)$ can be written as follows

$$p(z_i = 0|\pi, \mu, x_i) = (1 - \pi)\mathcal{N}(x_i|\mu_0, 0.25) = \alpha_0 \quad (5)$$

$$p(z_i = 1|\pi, \mu, x_i) = \pi\mathcal{N}(x_i|\mu_1, 0.25) = \alpha_1 \quad (6)$$

$$p(z_i|\pi, \mu, x_i) \sim Bernoulli(\frac{\alpha_1}{\alpha_0 + \alpha_1}) \quad (7)$$

; the variable π is conditional independent from μ and x given z, then the full conditional is

$$p(\pi|z, \mu, x) = Beta(0.8 + N_1, 0.8 + N_0) \quad (8)$$

$$N_0 = \sum_{i=1, z_i=0}^{100} 1,\ N_1 = \sum_{i=1, z_i=1}^{100} 1 \quad (9)$$

; and μ_i is independent from μ_{-i} and π given z and x, then the full conditional is

$$\sigma_0^2 = \left(\frac{1}{5^2} + \frac{N_0}{0.25^2}\right)^{-1},\ \sigma_1^2 = \left(\frac{1}{5^2} + \frac{N_1}{0.25^2}\right)^{-1} \quad (10)$$

$$m_0 = \sigma_0^2\left(\frac{\sum_{i=1, z_i=0}^{100} x_i}{0.25^2}\right),\ m_1 = \sigma_1^2\left(\frac{\sum_{i=1, z_i=1}^{100} x_i}{0.25^2}\right) \quad (11)$$

$$p(\mu_0|z, x) \sim \mathcal{N}(m_0, \sigma_0),\ p(\mu_1|z, x) \sim \mathcal{N}(m_1, \sigma_1) \quad (12)$$

With all previous full conditionals we can implement the Gibbs sampling algorithm.

Finally, the results is illustrated in Figure 4-5, where we can observe that the algorithms fastly converge to a negihborhood of the target parameters used to draw the data. The expected value approximation, i.e., the sample mean, are $\pi = 0.770$, $\mu_0 = 2.275$ and $\mu_1 = -5.954$, very close to the original values. To sum up, the Gibbs Sampling algorithm is an excellent algorithm to approximate complicated densities with a directed graph structure that encodes the conditional indepedencies in the model.

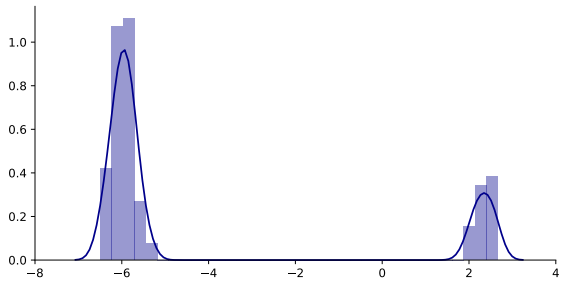

Figure 3: Random sample of a mixture of two Gaussians.

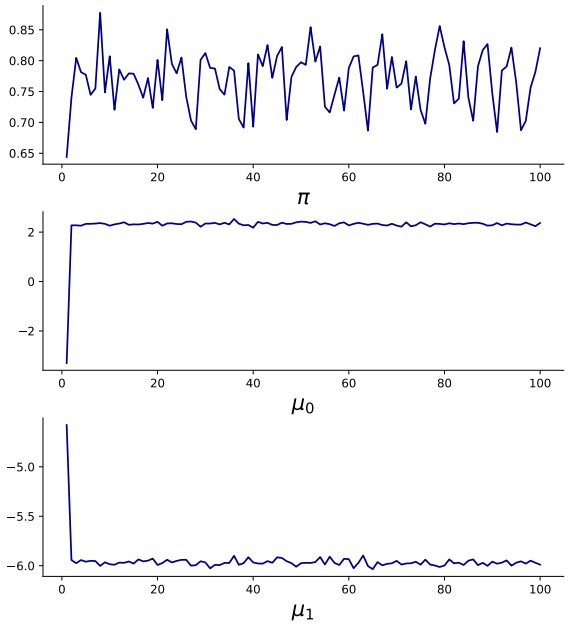

Figure 4: Posterior samples drawn from the Gibbs Sampling algorithm.

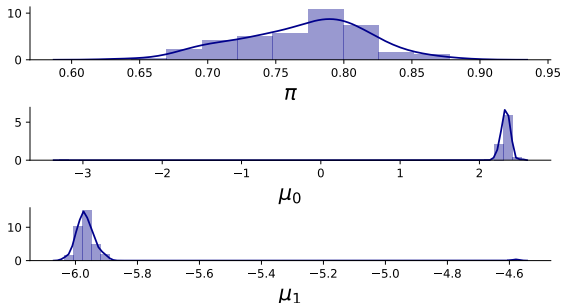

Figure 5: Posterior approximation from the Gibbs Sampling algorithm.