CIS 5200: Machine Learning	Spring 2023

Lecture 13 — February 28, 2023

Lecturer: Eric Wong *Lecture Notes Scribed by: Keshav Ramji*

This lecture was based off David Blei's notes on Bayesian mixture models and Gibbs sampling.[1]

1.1 Starting from Gaussian Mixture Models

1.1.1 GMM Formulation

Recall the general setup for a Gaussian Mixture Model (GMM) for sample x and cluster (Gaussian) z:

$$p(x, z) = p(x|z)p(z) = \phi_z(x)\pi(z) \tag{1.1}$$

In general, we can view this through the following **graphical model** structure: for the m samples, we have $z_i \to x_i$, where we have π (the distribution over the latent variables) as an input for the z_i (i.e. $z_i \sim \text{CAT}(\pi)$), and for each of the k clusters/Gaussians, we have inputs μ_k, and σ for the x_i, such that the x_i is distributed $N(\mu_{z_i}, \sigma^2 I)$.

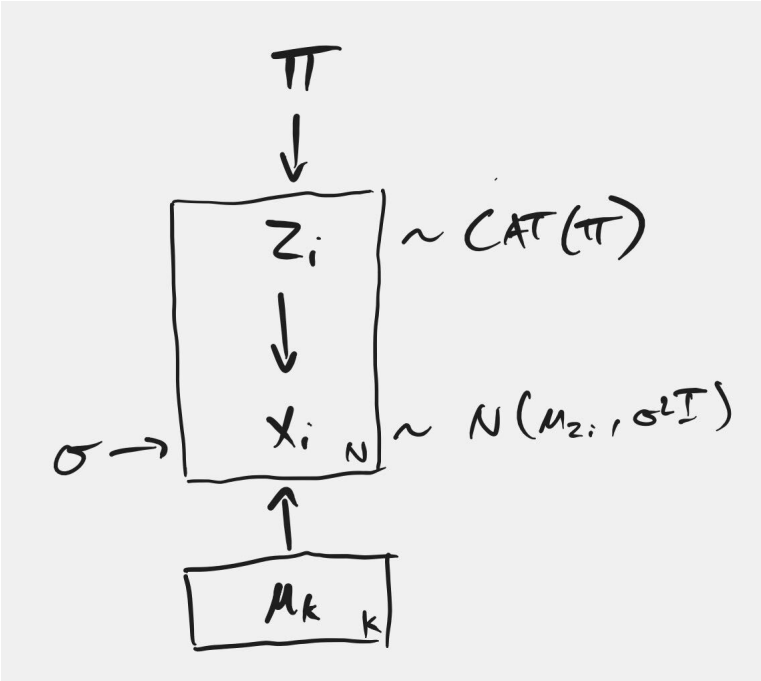

Figure 1.1. Graphical model of the GMM.

[1] http://www.cs.columbia.edu/~blei/fogm/2015F/notes/mixtures-and-gibbs.pdf

1.1.2 Bayesian Gausian Mixture Model (GMM)

Consider the above framework. Introduce a new parameter λ, which refers to the standard deviation for the prior for the centers of the clusters. That is, λ is an input to μ_k for the k clusters, such that the prior distribution for the μ_k is $N(0, \lambda^2 I)$. Essentially, the key difference is that μ_k is now a random variable, with an input parameter λ, which follows a Gaussian distribution with mean 0 and variance $\lambda^2 I$.

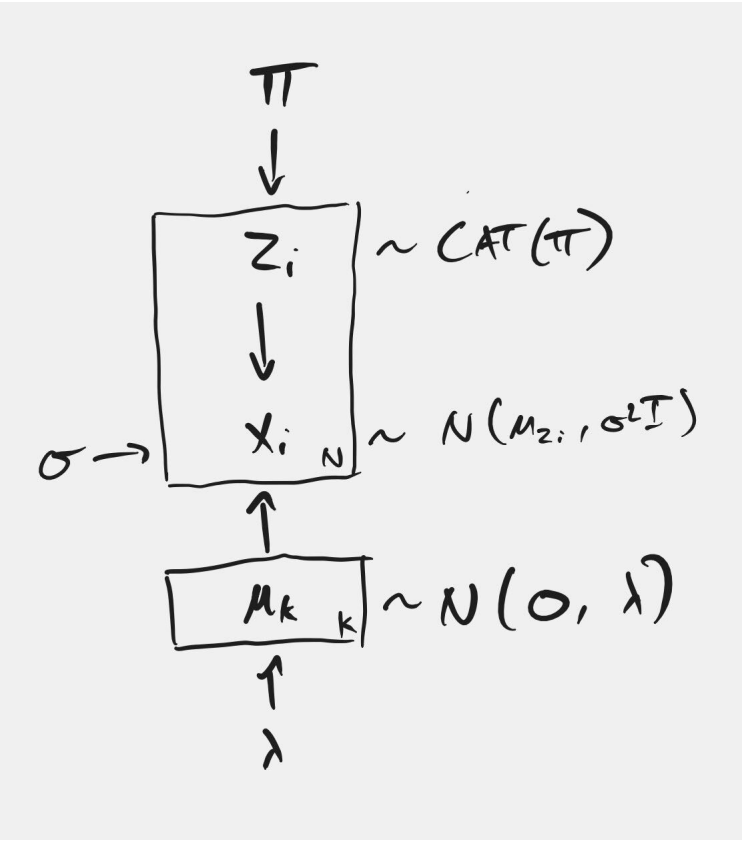

Figure 1.2. Graphical model of the Bayesian GMM with a prior on μ_k.

Consider the following real-world application: let x_i (our observations) represent a biomarker for the i^{th} patient, and the z_i (latent variables), represent subpopulations (e.g. deifned by age, race, gender, etc.). We can structure this as a GMM with k clusters – the question at hand is: how likely is μ_k, given the observations? That is, we want to estimate $P(\mu_k|X)$. Firstly, observe that this is not the same as $\pi_k = P(Z = k)$, as π controls the relative size of each cluster but not the cluster locations. By Bayes' Rule and conditional probability:

$$P(\mu_k|X) = \frac{P(X|\mu_k) * P(\mu_k)}{P(X)} \tag{1.2}$$

$$P(\mu_k|X) \propto P(X|\mu_k) * P(\mu_k) \tag{1.3}$$

Remark: $P(\mu_k|X)$ is often referred to as the posterior distribution. The ability to have a posterior is exactly why one might want to "go Bayesian" and introduce a prior on μ_k.

The posterior lets you answer a number of questions that the original setting cannot answer, since the original GMM setting has $P(\mu_k|X) = \delta_{\mu_k}$. For example, one cannot answer the question of "how likely" is any particular cluster centroid μ_k in the original GMM setting. One also cannot quantify the degree of uncertainty or confidence intervals around estimates of μ_k in the original GMM setting.

1.1.3 An Aside: Probabilistic Graphical Models (PGMs)

An aside: Probabilistic Graphical Models (PGMs) are a field of ML research focused on defining such graphical model structures. In other words, both the GMM and Bayesian GMM structures are PGMs. In this setting, we define the observations (i.e. x_i) as nodes, often in a directed graph structure, such that conditional probabilities can be used to estimate the joint distribution over the observations.

1.2 Building up to MCMC

Ultimately, we can view this by 2 key components of machine learning:

Learning

The learning problem is, given the data X, to estimate the parameters the model. Our goal, in the setting of the Bayesian GMM, for example, is to estimate certain parameters, e.g. π, σ, and λ. This can be done through methods such as the Expectation Maximization (EM) algorithm. These extensions of the EM algorithm for Bayesian models falls under the category known as "Variational Inference", or Variational Bayesian Methods.

Inference

The inference problem is, given the parameters of a model, to make new predictions at test time. That is, what is of interest to us is $P(X_{\mu+1}|X)$. In other words, this is the probability of seeing a new data point, at test time, given the data that we've seen so far. In a GMM, this evaluates to the following expression:

$$P(X_{m+1}|X) = \sum_k \phi_k(X_{m+1}) * \pi_k \tag{1.4}$$

In a Bayesian GMM, the expression evaluates to the following:

$$P(X_{m+1}|X) = \sum_k P(X_{m+1}|X, z = k)p(z = k) \tag{1.5}$$

$$= \sum_k P(z = k) * \int_{\mu_k} P(X_{m+1}|\mu_k) * P(\mu_k|X)d\mu \tag{1.6}$$

$$= \sum_k P(z = k) * E_{\mu|X}[P(X_{m+1}|\mu)] \tag{1.7}$$

However, evaluating this integral exactly at inference time is challenging! Hence, we now turn to approximation to solve this problem.

1.3 Markov Chain Monte-Carlo (MCMC) Methods)

Markov Chain Monte-Carlo (MCMC) Methods solve this problem by approximating this integral using samples – that is, using the principle of Monte Carlo simulation (running many samples over instances of the data and taking the expectation over the outcome). We can compute the expectation over $\mu|X$ as follows, for B simulations $\mu^{(1)}, \ldots, \mu^{(B)}$:

$$E_{\mu|X}[P(X_{m+1}|\mu)] = \frac{1}{B}\sum_{b} P(X_{m+1}|\mu^{(b)}) \tag{1.8}$$

Now we have started by applying the Monte Carlo principle, what remains is to draw samples from $p(\mu|X)$. We will use Markov Chains to draw samples from $\mu^{(b)}$, to compute this expectation.

1.3.1 Markov Chains

The idea of applying Markov Chains here is as follows: we will construct a sequence of $\mu^{(i)}$ (i.e. $\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(n)}$) that has the following property:

$$\lim_{n\to\infty} \mu^{(n)} \sim P(\mu|X) \tag{1.9}$$

We won't go into the specifics, but the sequence will satisfy the Markov Property:

$$P(\mu^{(t+1)}|X, \mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(t)}) = P(\mu^{(t+1)}|\mu^{(t)}) \tag{1.10}$$

Sequences satisfying the Markov property (and some other conditions) are known to have a limiting distribution as n tends to infinity. The trick is to create a sequence whose limiting distribution matches the target distribution of interest. This idea is the basis of MCMC algorithms. Gibbs Sampling, which we'll discuss next, is one way to create such a sequence such that the limiting distribution is exactly $P(\mu|X)$.

When the limiting distribution matches the target distribution $P(\mu|X)$, then there is a natural way to draw samples from the target distribution: simply generate a very long sequence and then take the last sample. This gets you one sample from the target distribution. Repeating this B times will result in a batch of B samples.

1.4 Gibbs Sampling

Gibbs Sampling is one of the simpler sampling algorithms from the family of MCMC algorithms. As inputs, we have:

- $X = \{x_1, x_2, \ldots, x_m\}$
- The parameters of the Bayesian GMM algorithm, namely π, λ and σ.
- Initialize μ_k, z_i for all K components and M examples

The unknowns, or the latent variables, in the Bayesian GMM are the μ_k and z_i variables. The idea of Gibbs sampling is to iterate over all latent variables and sample a new latent variable conditioned on current value of all other latent variables. This is shown in Algorithm 1.

Algorithm 1 Gibbs Sampling

for i = 1 ... M **do**
  Sample $z_i \sim P(z_i|\mu, (Z \setminus z_i), X)$
end for
for k = 1 ... K **do**
  Sample $\mu_k \sim P(\mu_k|Z, (\mu \setminus \mu_k), X)$
end for

How do we sample from these conditionals? The core assumption in Gibbs sampling is that the conditionals are easy to sample from. For the Bayesian GMM, this is the case. Observe the following simplification of the probabilities of interest in the Gibbs Sampling Algorithm:

$$P(z_i|\mu, z_i, X) \propto P(x_i|(Z \setminus z_i), \mu) * P(z_i) \tag{1.11}$$

$$= \frac{\phi_{z_i}(x_i) * \pi_k}{P(\mu_k|Z, (\mu \setminus \mu_k), X)} \tag{1.12}$$

$$\propto \phi_{z_i}(x_i) * \pi_k \tag{1.13}$$

$$\tag{1.14}$$

In other words, $P(z_i|\mu, z_i, X)$ is simply a categorical distribution where each cluster has probability $\phi_{z_i}(x_i) * \pi_k$ of being chosen. This can be sampled from easily! How do we sample from $P(\mu_k|Z, (\mu \setminus \mu_k), X)$?

$$P(\mu_k|Z, (\mu \setminus \mu_k), X) \tag{1.15}$$

$$= P(\mu_k|Z, X) \tag{1.16}$$

$$\propto P(X|Z, \mu_k) * P(\mu_k) \tag{1.17}$$

There is a well known algebraic result here that says that, since $P(X|Z, \mu_k)$ is Gaussian and $P(\mu_k)$ is Gaussian, then the resulting distribution $P(\mu_k|Z, X)$ is also Gaussian with known parameters. This result is also sometimes referred to as a *conjguate prior*.[2] Specifically, this is

$$P(\mu_k|Z, X) = N(\hat{\mu}_k, \hat{\lambda}_k) \tag{1.18}$$

where

$$\hat{\mu}_k = \left(\frac{n_k/\sigma^2}{n_k/\sigma^2 + 1/\lambda^2}\right) \bar{X}_k \tag{1.19}$$

$$\hat{\lambda}_k = (n_k/\sigma^2 + 1/\lambda^2)^I - 1 \tag{1.20}$$

[2] https://en.wikipedia.org/wiki/Conjugate_prior

and

$$n_k = \sum_{i=1}^{m} 1(z_i = k) \tag{1.21}$$

$$\hat{\lambda}_k = \frac{\sum_{i=1}^{m} 1(z_i = k) * x_i}{n_k} \tag{1.22}$$

Remark: Gibbs sampling is just one technique in the broader family of MCMC sampling methods. There are in fact other ways to construct sequences whose limit is the posterior distribution. Gibbs sampling can be viewed as a special case of the Metropolis-Hastings sampling technique, which uses samples from a proposal distribution to generate the Markov chain. The relation here is that Gibbs sampling is a Metropolis-Hastings algorithm where the proposal distribution is exactly the conditional distribution of one variable conditioned on all the others.