Metropolis-Hastings; Introduction to finite mixture models

Dr. Olanrewaju Michael Akande

April 10, 2020

Announcements

- Deadline for requesting change of grade is now April 22 (see https://registrar.duke.edu/forms/spring-2020-grading-basis-change-graded).
- Details on final exam next week; review session next Wednesday.

Outline

- Metropolis-Hastings algorithm
  - Introduction and intuition
  - Algorithm
- Finite mixture models
  - Categorical data – univariate case
  - Continuous data – univariate case

Metropolis-Hastings algorithm

Metropolis-Hastings algorithm

- Gibbs sampling and the Metropolis algorithm are special cases of the Metropolis-Hastings algorithm.
- The Metropolis-Hastings algorithm is more general in that it allows arbitrary proposal distributions.
- These can be symmetric around the current values, full conditionals, or something else entirely as long as they do not depend on values in our sequence that are previous to the most current values.
- That last point is to ensure the sequence is a Markov chain!
- In terms of how this works, the only real change from before is that now, the acceptance probability should also incorporate the proposal density when it is not symmetric.

Metropolis-Hastings algorithm

- Suppose our target distribution is $p_0(\theta)$. The algorithm proceeds as follows:

  1. Given a current draw $\theta^{(s)}$, propose a new value $\theta^\star \sim g_\theta[\theta^\star|\theta^{(s)}]$.
  2. Compute the acceptance ratio

$$r = \frac{p_0(\theta^\star)}{p_0(\theta^{(s)})} \times \frac{g_\theta[\theta^{(s)}|\theta^\star]}{g_\theta[\theta^\star|\theta^{(s)}]}.$$

  3. Sample $u \sim U(0,1)$ and set

$$\theta^{(s+1)} = \begin{cases} \theta^\star & \text{if} \quad u < r \\ \theta^{(s)} & \text{if} \quad \text{otherwise} \end{cases}.$$

Metropolis-Hastings algorithm

- If $p_0(\theta)$ corresponds to a posterior distribution $\pi(\theta|y)$ as is often the case for us, then we have

1. Propose a new value $\theta^\star \sim g_\theta[\theta^\star|\theta^{(s)}]$.

2. Compute the acceptance ratio

$$\begin{aligned} r &= \frac{\pi(\theta^\star|y)}{\pi(\theta^{(s)}|y)} \times \frac{g_\theta[\theta^{(s)}|\theta^\star]}{g_\theta[\theta^\star|\theta^{(s)}]} \\ &= \frac{\mathcal{L}(y|\theta^\star)\pi(\theta^\star)}{\mathcal{L}(y|\theta^{(s)})\pi(\theta^{(s)})} \times \frac{g_\theta[\theta^{(s)}|\theta^\star]}{g_\theta[\theta^\star|\theta^{(s)}]}. \end{aligned}$$

3. Sample $u \sim U(0,1)$ and set

$$\theta^{(s+1)} = \begin{cases} \theta^\star & \text{if} \quad u < r \\ \theta^{(s)} & \text{if} \quad \text{otherwise} \end{cases}.$$

Metropolis-Hastings algorithm

- Suppose our target distribution is $p_0(u, v)$, a bivariate distribution for random variables U and V.
- For example, $p_0(u, v)$ could be the joint posterior distribution for U and V.
- Two options:
  - Define one joint proposal density $g_{u,v}[u^\star, v^\star | u^{(s)}, v^{(s)}]$ for U and V if possible; or
  - Define two proposal densities, one for U and the other for V. That is, $g_u[u^\star | u^{(s)}, v^{(s)}]$ and $g_u[v^\star | u^{(s)}, v^{(s)}]$.
- First option follows directly from the main algorithm and often works very well when possible. Second option needs a little modification.

Metropolis-Hastings algorithm

1. Update U: first, sample $u^\star \sim g_u[u^\star|u^{(s)}, v^{(s)}]$. Then,
   - Compute the acceptance ratio
   $$r = \frac{p_0(u^\star, v^{(s)})}{p_0(u^{(s)}, v^{(s)})} \times \frac{g_u[u^{(s)}|u^\star, v^{(s)}]}{g_u[u^\star|u^{(s)}, v^{(s)}]}.$$
   - Sample $w \sim U(0,1)$. Set $u^{(s+1)}$ to $u^\star$ if $w < r$, or set $u^{(s+1)}$ to $u^\star$ otherwise.
2. Update V: first sample $v^\star \sim g_v[v^\star|u^{(s+1)}, v^{(s)}]$. Then,
   - Compute the acceptance ratio
   $$r = \frac{p_0(u^{(s+1)}, v^\star)}{p_0(u^{(s+1)}, v^{(s)})} \times \frac{g_v[v^{(s)}|u^{(s+1)}, v^\star]}{g_v[v^\star|u^{(s+1)}, v^{(s)}]}.$$
   - Sample $w \sim U(0,1)$. Set $v^{(s+1)}$ to $v^\star$ if $w < r$, or set $v^{(s+1)}$ to $v^\star$ otherwise.

Metropolis-Hastings algorithm

- The acceptance ratio looks like what we had before except with an additional factor.
- That factor is the ratio of the probability of generating the current value from the proposed to the probability of generating the proposed value from the current (ratio is equal to one for symmetric proposal – see homework!).
- Also, it is often the case that full conditionals are available for some parameters but not all.
- Very useful trick is to combine Gibbs and Metropolis.
- Unfortunately, we do not have enough time to get into examples on Metropolis-Hastings or how combine Gibbs and Metropolis.
- Please read Chapter 10.5 of the Hoff book to see how it works!!

Finite mixture models

Categorical data – univariate case

- We begin our development of finite mixture models by going back to the Dirichlet-multinomial model.
- First, recall that if $y_i, \ldots, y_n | \boldsymbol{\theta} \overset{iid}{\sim} \text{Categorical}(\boldsymbol{\theta})$, then

$$\Pr[y_i = d | \boldsymbol{\theta}] = \prod_{d=1}^{D} \theta_d^{1[y_i = d]},$$

where

  - $y_i \in \{1, \ldots, D\}$,
  - $\Pr(y_i = d) = \theta_d$ for each $d = 1, \ldots, D$, and
  - $\sum_{d=1}^{D} \theta_d = 1, \;\; \theta_d \geq 0 \;\; \text{for all} \;\; d = 1, \ldots, D.$

- So that

$$L[Y|\boldsymbol{\theta}] = \prod_{i=1}^{n} \prod_{d=1}^{D} \theta_d^{1[y_i = d]} = \prod_{d=1}^{D} \theta_d^{\sum_{i=1}^{n} 1[y_i = d]} = \prod_{d=1}^{D} \theta_d^{n_d}$$

where n_d is just the number of observations in category d.

Categorical data – univariate case

- Then, a conjugate prior for categorical/multinomial data is the Dirichlet distribution.
- With prior $\pi[\boldsymbol{\theta}] = \text{Dirichlet}(\boldsymbol{\alpha})$, we have

$$\pi[\boldsymbol{\theta}] \propto \prod_{d=1}^{D} \theta_j^{\alpha_j - 1}, \quad \alpha_j > 0 \text{ for all } d = 1, \ldots, D.$$

  where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$.

- So that the posterior is

$$\pi(\boldsymbol{\theta}|Y) = \text{Dirichlet}(\alpha_1 + n_1, \ldots, \alpha_d + n_d)$$

- However, what if our data actually comes from K different sub-populations of groups of people?
- For example, if our data comes from men and women, and we don't expect marginal independence across the two groups (vote turnout, income, etc), then we have a mixture of distributions.

Categorical data – univariate case

- With our data coming from a "combination" or "mixture" of sub-populations, we no longer have independence across all observations, so that the likelihood $L[Y|\boldsymbol{\theta}] \neq \prod_{i=1}^{n} \prod_{d=1}^{D} \theta_j^{1[y_i=d]}$.
- However, we can still have "conditional independence" within each group.
- Unfortunately, we do not always know the indexes for those groups.
- That is, we know our data contains K different groups, but we actually do not know which observations belong to which groups.
- **Solution**: introduce a latent variable z_i representing the group/cluster indicator for each observation i, so that each $z_i \in \{1, \ldots, K\}$.
- This is a form of data augmentation, but we will define that properly later.

Finite mixture of multinomials

- Given the cluster indicator z_i for observation i, write
  - $\Pr(y_i = d|z_i) = \psi_{z_i,d} \equiv \prod_{d=1}^{D} \psi_{z_i,d}^{1[y_i=d|z_i]}$, and
  - $\Pr(z_i = k) = \lambda_k \equiv \prod_{k=1}^{K} \lambda_k^{1[z_i=k]}$.
- Then, the marginal probabilities we care about will be

$$\begin{aligned}\theta_d &= \Pr(y_i = d) \\ &= \sum_{k=1}^{K} \Pr(y_i = d|z_i = k) \cdot \Pr(z_i = k) \\ &= \sum_{k=1}^{K} \lambda_k \cdot \psi_{z_i,d},\end{aligned}$$

which is a finite mixture of multinomials, with the weights given by λ_k.

Posterior inference

- Write
  - $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$, and
  - $\boldsymbol{\psi} = \{\psi_{z_i,d}\}$ to be a $K \times D$ matrix of probabilities, where each kth row is the vector of probabilities for cluster k.
- The observed data likelihood is

$$L\left[Y = (y_1, \ldots, y_n) | Z = (z_1, \ldots, z_n), \boldsymbol{\psi}, \boldsymbol{\lambda}\right] = \prod_{i=1}^{n} \prod_{d=1}^{D} \Pr\left(y_i = d | z_i, \psi_{z_i,d}\right)$$
$$= \prod_{i=1}^{n} \prod_{d=1}^{D} \psi_{z_i,d}^{1[y_i=d|z_i]},$$

  which includes products (and not the sums in the mixture pdf), and as you will see, makes sampling a bit easier.
- Next we need priors.

Posterior inference

- First, for $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$, the vector of cluster probabilities, we can use a Dirichlet prior. That is,

$$\pi[\boldsymbol{\lambda}] = \text{Dirichlet}(\alpha_1, \ldots, \alpha_K) \propto \prod_{k=1}^{K} \lambda_k^{\alpha_k - 1}.$$

- For ψ, we can assume independent Dirichlet priors for each cluster vector $\boldsymbol{\psi}_k = (\psi_{k,1}, \ldots, \psi_{k,D})$. That is, for each $k = 1, \ldots, K$,

$$\pi[\boldsymbol{\psi}_k] = \text{Dirichlet}(a_1, \ldots, a_d) \propto \prod_{d=1}^{D} \psi_{k,d}^{a_d - 1}.$$

- Finally, from our distribution on the z_i's, we have

$$\Pr\left[Z = (z_1, \ldots, z_n) | \boldsymbol{\lambda}\right] = \prod_{i=1}^{n} \prod_{k=1}^{K} \lambda_k^{1[z_i = k]}.$$

Posterior inference

- Note that the unobserved variables and parameters are $Z = (z_1, \ldots, z_n)$, $\boldsymbol{\psi}$, and $\boldsymbol{\lambda}$.
- So, the joint posterior is

$$\begin{aligned}
\pi(Z, \boldsymbol{\psi}, \boldsymbol{\lambda} | Y) &\propto L[Y | Z, \boldsymbol{\psi}, \boldsymbol{\lambda}] \cdot \Pr(Z | \boldsymbol{\psi}, \boldsymbol{\lambda}) \cdot \pi(\boldsymbol{\psi}, \boldsymbol{\lambda}) \\
&\propto \left[\prod_{i=1}^{n} \prod_{d=1}^{D} \Pr\left(y_i = d | z_i, \psi_{z_i, d}\right) \right] \cdot \Pr(Z | \boldsymbol{\lambda}) \cdot \pi(\boldsymbol{\psi}) \cdot \pi(\boldsymbol{\lambda}) \\
&\propto \left(\prod_{i=1}^{n} \prod_{d=1}^{D} \psi_{z_i, d}^{1[y_i = d | z_i]} \right) \\
&\quad \times \left(\prod_{i=1}^{n} \prod_{k=1}^{K} \lambda_k^{1[z_i = k]} \right) \\
&\quad \times \left(\prod_{k=1}^{K} \prod_{d=1}^{D} \psi_{k,d}^{a_d - 1} \right) \\
&\quad \times \left(\prod_{k=1}^{K} \lambda_k^{\alpha_k - 1} \right).
\end{aligned}$$

Posterior inference

- First, we need to sample the z_i's, one at a time, from their full conditionals.
- For $i = 1, \ldots, n$, sample $z_i \in \{1, \ldots, K\}$ from a categorical distribution (multinomial distribution with sample size one) with probabilities

$$
\begin{aligned}
\Pr[z_i = k | \ldots] &= \Pr[z_i = k | y_i, \boldsymbol{\psi}_k, \lambda_k] \\
&= \frac{\Pr[y_i, z_i = k | \boldsymbol{\psi}_k, \lambda_k]}{\sum_{l=1}^{K} \Pr[y_i, z_i = l | \boldsymbol{\psi}_l, \lambda_l]} \\
&= \frac{\Pr[y_i | z_i = k, \boldsymbol{\psi}_k] \cdot \Pr[z_i = k, \lambda_k]}{\sum_{l=1}^{K} \Pr[y_i | z_i = l, \boldsymbol{\psi}_l] \cdot \Pr[z_i = l, \lambda_l]} \\
&= \frac{\psi_{k,d} \cdot \lambda_k}{\sum_{l=1}^{K} \psi_{l,d} \cdot \lambda_l}.
\end{aligned}
$$

Posterior inference

- Next, sample each cluster vector $\boldsymbol{\psi}_k = (\psi_{k,1}, \ldots, \psi_{k,D})$ from

$$
\begin{aligned}
\pi[\boldsymbol{\psi}_k | \ldots] &\propto \pi(Z, \boldsymbol{\psi}, \boldsymbol{\lambda} | Y) \\
&\propto \left(\prod_{i=1}^{n} \prod_{d=1}^{D} \psi_{z_i,d}^{1[y_i=d|z_i]} \right) \cdot \left(\prod_{i=1}^{n} \prod_{k=1}^{K} \lambda_k^{1[z_i=k]} \right) \cdot \left(\prod_{k=1}^{K} \prod_{d=1}^{D} \psi_{k,d}^{a_d-1} \right) \cdot \left(\prod_{k=1}^{K} \lambda_k^{\alpha_k-1} \right) \\
&\propto \left(\prod_{d=1}^{D} \psi_{k,d}^{n_{k,d}} \right) \cdot \left(\prod_{d=1}^{D} \psi_{k,d}^{a_d-1} \right) \\
&= \left(\prod_{d=1}^{D} \psi_{k,d}^{a_d+n_{k,d}-1} \right) \\
&\equiv \text{Dirichlet}\left(a_1 + n_{k,1}, \ldots, a_d + n_{k,D}\right).
\end{aligned}
$$

where $n_{k,d} = \sum_{i:z_i=k} 1[y_i = d]$, the number of individuals in cluster k that are assigned to category d of the levels of y.

Posterior inference

- Finally, sample $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$, the vector of cluster probabilities from

$$
\begin{aligned}
\pi[\boldsymbol{\lambda}|\ldots] &\propto \pi\left(Z, \boldsymbol{\psi}, \boldsymbol{\lambda}|Y\right) \\
&\propto \left(\prod_{i=1}^{n}\prod_{d=1}^{D}\psi_{z_i,d}^{1[y_i=d|z_i]}\right)\cdot\left(\prod_{i=1}^{n}\prod_{k=1}^{K}\lambda_k^{1[z_i=k]}\right)\cdot\left(\prod_{k=1}^{K}\prod_{d=1}^{D}\psi_{k,d}^{a_d-1}\right)\cdot\left(\prod_{k=1}^{K}\lambda_k^{\alpha_k-1}\right) \\
&\propto \left(\prod_{i=1}^{n}\prod_{k=1}^{K}\lambda_k^{1[z_i=k]}\right)\cdot\left(\prod_{k=1}^{K}\lambda_k^{\alpha_k-1}\right) \\
&\propto \left(\prod_{k=1}^{K}\lambda_k^{n_k}\right)\cdot\left(\prod_{k=1}^{K}\lambda_k^{\alpha_k-1}\right) \\
&\propto \left(\prod_{k=1}^{K}\lambda_k^{\alpha_k+n_k-1}\right) \\
&\equiv \text{Dirichlet}\left(a_1+n_1, \ldots, a_d+n_d\right),
\end{aligned}
$$

where $n_k = \sum_{i=1}^{n} 1[z_i = k]$, the number of individuals assigned to cluster k.

Continuous data – univariate case

- What about continuous data? Suppose we have univariate data $y_i \overset{iid}{\sim} f$, for $i, \ldots, n$, where f is an unknown density.
- Turns out that we can approximate "almost" any f with a mixture of normals. Usual choices are

1. Location mixture (multimodal):

$$f(y) = \sum_{k=1}^{K} \lambda_k \mathcal{N}\left(\mu_k, \sigma^2\right)$$

2. Scale mixture (unimodal and symmetric about the mean, but fatter tails than a regular normal distribution):

$$f(y) = \sum_{k=1}^{K} \lambda_k \mathcal{N}\left(\mu, \sigma_k^2\right)$$

3. Location-scale mixture (multimodal with potentially fat tails):

$$f(y) = \sum_{k=1}^{K} \lambda_k \mathcal{N}\left(\mu_k, \sigma_k^2\right)$$

Location mixture of normals

- Consider the location mixture $f(y) = \sum_{k=1}^{K} \lambda_k \mathcal{N}\left(\mu_k, \sigma^2\right)$. How can we do inference?
- Right now, we only have three unknowns: $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, and σ^2.
- For priors, the most obvious choices are
  - $\pi[\boldsymbol{\lambda}] = \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$,
  - $\mu_k \sim \mathcal{N}(\mu_0, \gamma_0^2)$, for each $k = 1, \ldots, K$, and
  - $\sigma^2 \sim \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2}\right)$.
- However, we do not want to use the likelihood with the sum in the mixture. We prefer products!

Data augmentation

- This brings us the to concept of data augmentation, which we actually already used in the mixture of multinomials.
- Data augmentation is a commonly-used technique for designing MCMC samplers using auxiliary/latent/hidden variables. Again, we have already seen this.
- Idea: introduce variable(s) Z that depends on the distribution of the existing variables in such a way that the resulting conditional distributions, with Z included, are easier to sample from and/or result in better mixing.
- Z's are just latent/hidden variables that are introduced for the purpose of simplifying/improving the sampler.

Data augmentation

- For example, suppose we want to sample from $p(x, y)$, but $p(x|y)$ and/or $p(y|x)$ are complicated.
- Choose $p(z|x, y)$ such that $p(x|y, z)$, $p(y|x, z)$, and $p(z|x, y)$ are easy to sample from. Note that we have $p(x, y, z) = p(z|x, y)p(x, y)$.
- Alternatively, rewrite the model as $p(x, y|z)$ and specify $p(z)$ such that

$$p(x, y) = \int p(x, y|z)p(z)\mathrm{d}z,$$

  where the resulting $p(x|y, z)$, $p(y|x, z)$, and $p(z|x, y)$ from the joint $p(x, y, z)$ are again easy to sample from.
- Next, construct a Gibbs sampler to sample all three variables (X, Y, Z) from $p(x, y, z)$.
- Finally, throw away the sampled Z's and from what we know about Gibbs sampling, the samples (X, Y) are from the desired $p(x, y)$.

Location mixture of normals

- Back to location mixture $f(y) = \sum_{k=1}^{K} \lambda_k \mathcal{N}\left(\mu_k, \sigma^2\right)$.
- Introduce latent variable $z_i \in \{1, \ldots, K\}$.
- Then, we have
  - $y_i | z_i \sim \mathcal{N}\left(\mu_{z_i}, \sigma^2\right)$, and
  - $\Pr(z_i = k) = \lambda_k \equiv \prod_{k=1}^{K} \lambda_k^{1[z_i = k]}$.
- How does that help? Well, the observed data likelihood is now

$$L\left[Y = (y_1, \ldots, y_n) | Z = (z_1, \ldots, z_n), \boldsymbol{\psi}, \boldsymbol{\mu}, \sigma^2\right] = \prod_{i=1}^{n} p\left(y_i | z_i, \mu_{z_i}, \sigma^2\right)$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \mu_{z_i})^2\right\}$$

which is much easier to work with.

Posterior inference

- The joint posterior is

$$
\begin{aligned}
\pi\left(Z, \boldsymbol{\mu}, \sigma^2, \boldsymbol{\lambda} | Y\right) &\propto \left[\prod_{i=1}^{n} p\left(y_i | z_i, \mu_{z_i}, \sigma^2\right)\right] \cdot \Pr(Z | \boldsymbol{\mu}, \sigma^2, \boldsymbol{\lambda}) \cdot \pi(\boldsymbol{\mu}, \sigma^2, \boldsymbol{\lambda}) \\
&\propto \left[\prod_{i=1}^{n} p\left(y_i | z_i, \mu_{z_i}, \sigma^2\right)\right] \cdot \Pr(Z | \boldsymbol{\lambda}) \cdot \pi(\boldsymbol{\lambda}) \cdot \pi(\boldsymbol{\mu}) \cdot \pi(\sigma^2) \\
&\propto \left[\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \mu_{z_i})^2\right\}\right] \\
&\quad \times \left[\prod_{i=1}^{n} \prod_{k=1}^{K} \lambda_k^{1[z_i = k]}\right] \\
&\quad \times \left[\prod_{k=1}^{K} \lambda_k^{\alpha_k - 1}\right] \cdot \\
&\quad \times \left[\prod_{k=1}^{K} \mathcal{N}(\mu_k; \mu_0, \gamma_0^2)\right] \\
&\quad \times \left[\mathcal{IG}\left(\sigma^2; \frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2}\right)\right].
\end{aligned}
$$

Full conditionals

- For $i = 1, \ldots, n$, sample $z_i \in \{1, \ldots, K\}$ from a categorical distribution (multinomial distribution with sample size one) with probabilities

$$\begin{aligned}
\Pr[z_i = k | \ldots] &= \frac{\Pr[y_i, z_i = k | \mu_k, \sigma^2, \lambda_k]}{\sum_{l=1}^{K} \Pr[y_i, z_i = l | \mu_l, \sigma^2, \lambda_l]} \\
&= \frac{\Pr[y_i | z_i = k, \mu_k, \sigma^2] \cdot \Pr[z_i = k, \lambda_k]}{\sum_{l=1}^{K} \Pr[y_i | z_i = l, \mu_l, \sigma^2] \cdot \Pr[z_i = l, \lambda_l]} \\
&= \frac{\lambda_k \cdot \mathcal{N}\left(y_i; \mu_k, \sigma^2\right)}{\sum_{l=1}^{K} \lambda_l \cdot \mathcal{N}\left(y_i; \mu_l, \sigma^2\right)}.
\end{aligned}$$

Full conditionals

- Next, sample $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$ from

$$\pi[\boldsymbol{\lambda}|\ldots] \equiv \text{Dirichlet}\left(a_1 + n_1, \ldots, a_d + n_d\right),$$

where $n_k = \sum_{i=1}^{n} 1[z_i = k]$, the number of individuals assigned to cluster k.

- Sample the mean μ_k for each cluster from

$$\pi[\mu_k|\ldots] \equiv \mathcal{N}(\mu_{k,n}, \gamma^2_{k,n});$$

$$\gamma^2_{k,n} = \frac{1}{\frac{n_k}{\sigma^2} + \frac{1}{\gamma_0^2}}; \qquad \mu_{k,n} = \gamma^2_{k,n}\left[\frac{n_k}{\sigma^2}\bar{y}_k + \frac{1}{\gamma_0^2}\mu_0\right],$$

- Finally, sample σ^2 from

$$\pi(\sigma^2|\ldots) = \mathcal{IG}\left(\frac{\nu_n}{2}, \frac{\nu_n\sigma_n^2}{2}\right).$$

$$\nu_n = \nu_0 + n; \qquad \sigma_n^2 = \frac{1}{\nu_n}\left[\nu_0\sigma_0^2 + \sum_{i=1}^{n}(y_i - \mu_{z_i})^2\right].$$

Inference

- We will take a quick look at an example in the last class!
- For categorical data with two or more categorical variables, it is relatively easy to extend the framework.
- If interested, read up on finite mixture of products of multinomials. Happy to provide resources for those interested.
- How to choose k, the number of clusters?
  - Compare marginal likelihood for different choices of k and select k with best performance.
  - Can also use other metrics MSE, and so on. Maybe even cross validation.
  - Go Bayesian non-parametric: Dirichlet processes!