SDS 383D: Modeling II **Spring 2017**

Section 5: Mixture models

Mixture models

So far, we've assumed that our data are conditionally exchangeable given their covariates. In other words, for every unique set of covariates there exists a set of parameters, conditioned on which, the data with those covariates are i.i.d. We used various distributions over functions to learn a distribution over these parameters, for all covariate settings.

A common setting was when our data was normally distributed, with mean $\beta^T x_i$ and variance σ^2. If we did not have the covariate values x_i, our data would no longer be normally distributed.

Exercise 5.1 *Download the dataset restaurants.csv. This contains profit information for restaurants, based on seating capacity and whether they are open for dinner. Run a Bayesian regression of Profit vs SeatingCapacity and a dummy for DinnerService (you can reuse code from 2.12) (I'd suggest whitening Profit, it will make later prior specification easier). Do the residuals look normal? (e.g. plot histograms, qq plots). Now, let's just look at the raw Profit data: Does it look normal?*

Solution: we know that

$$\beta|\omega, \mathbf{y} \sim \mathcal{N}(\mu_p, (\omega K_p)^{-1})$$

$$\omega|\mathbf{y} \sim Gamma(a + \frac{n}{2}, b + \frac{\mathbf{y}^T\Lambda\mathbf{y} + \mu^T K\mu - \mu_p^T K_p \mu_p}{2})$$

with $\mu_p = (X^T\Lambda X + K)^{-1}(X^T\Lambda\mathbf{y} + K\mu)$ and $K_p = X^T\Lambda X + K$. I set $\Lambda = I$, $K = I$, $a = 2$, and $b = 1$ for the Bayesian model and compare the coefficients with linear regression (as shown in Table 1). Also, we computed the Rooted Mean Squared Errors for both models. From Table 1, we can see that Bayesian model has slightly small errors. The residual plot in the left panel of Figure 1 looks normal, while the histogram plot of the response variable `Profit` looks more like a mixture of normals.

Model	Intercept (β_1)	DinnerService (β_2)	SeatingCapacity (β_3)	RMSE
Least Squares	32732.38	19773.96	103.21	6882.28
Bayesian	32016.81	19745.03	106.67	6707.56

Table 1: Estimated $\beta = (\beta_1, \beta_2, \beta_3)$ and Rooted Mean Squared Errors for different regression models

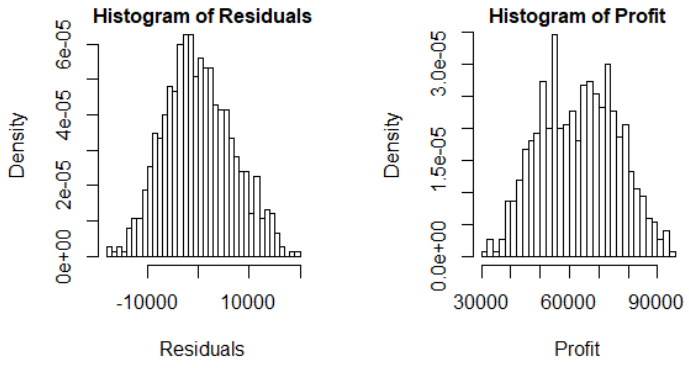


Figure 1: Histogram of residuals and response variable `Profit`

Let's assume we're in the situation where we don't know any of these covariate values. For now, let's ignore the continuous-valued covariate (SeatingCapacity), and try to infer the categorical covariate. Let's say we know that half our restaurants are open for dinner. We could assume that each restaurant is associated with a *latent* indicator variable Z_i, that assigns them to one of two groups, so that

$$Z_i \sim \text{Bernoulli}(\pi)$$

As in the regression setting, conditioned on the latent variable, we will assume that the observed profits are i.i.d. normal. Again, as in the basic regression setting, we will assume the variances of the two normals are the same, but the means are different, i.e.

$$X_i | Z_i = z \sim \text{Normal}(\mu_z, \sigma^2).$$

If we marginalize over these binary indicators, our observations are assumed to be distributed according to a mixture of two Gaussians:

$$X_i \sim 0.5N(\mu_0, \sigma_0^2) + 0.5(\mu_1, \sigma_1^2)$$

We can then look at the posterior distribution over each indicator variable, conditioned on the class probabilities and parameters:

$$\begin{aligned} & \mathbf{P}(Z_i = z | X_i, \pi, \mu_z, \sigma^2) \propto P(Z_i = z|\pi) p(X_i|\mu_z, \sigma^2) \\ \text{so,} \quad & \mathbf{P}(Z_i = 1 | X_i, \pi, \mu_1, \sigma^2) \propto P(Z_i = 1|\pi) p(X_i|\mu_1, \sigma^2) \\ & \mathbf{P}(Z_i = 0 | X_i, \pi, \mu_0, \sigma^2) \propto P(Z_i = 0|\pi) p(X_i|\mu_0, \sigma^2) \end{aligned}$$

Conditioned on the Z_i, we can update the means of the Gaussians using conjugacy.

Note that we are not guaranteed to find latent clusters that correspond to the covariate we were expecting! If there is a more parsimonious partitioning of the data, then the posterior will tend to favor that partitioning.

Exercise 5.2 *Let's assume (as is the case if our latent variables correspond to the actual DinnerService covariate) that the class proportions are roughly equal, and fix* $\pi = 0.5$. *Using the conditional distributions* $P(Z_i|X_i, \pi, \mu_1, \mu_2, \sigma^2)$ *and* $p(\mu_k|\{X_i : Z_i = k\}, \alpha)$, *where* α *are appropriate (shared) prior parameters for* μ_k *and* $k \in \{0, 1\}$. *Implement a Gibbs sampler that samples the means, variance, and the latent indicator variables by using the parameters of the initial regression to pick hyperparameters. Compare the clustering obtained with the "true" clustering due to the DinnerService variable.*

Solution: Denote $\mathbf{X} = (X_1, \ldots, X_n)$, $\theta = (\mu_0, \mu_1, \sigma^2)$, and $\mathbf{Z} = (Z_1, \ldots, Z_n)$, then the model is

$$f(X_i|\theta) = 0.5\mathcal{N}(X_i|\mu_0, \sigma^2) + 0.5\mathcal{N}(X_i|\mu_1, \sigma^2)$$

The prior of the parameters are

$$\begin{aligned} \mu_0 \sim & \mathcal{N}(m_0, 1/\lambda) \\ \mu_1 \sim & \mathcal{N}(m_1, 1/\lambda) \\ 1/\sigma^2 \sim & \text{Gamma}(a, b) \end{aligned}$$

Then the full likelihood is

$$f(\mathbf{X}|\theta) = \prod_{i=1}^{n}[0.5\mathcal{N}(X_i|\mu_0, \sigma^2) + 0.5\mathcal{N}(X_i|\mu_1, \sigma^2)]$$

The posterior distribution of θ is

$$p(\theta|\mathbf{X}) = f(\mathbf{X}|\theta)p(\theta) = \prod_{i=1}^{n}[0.5\mathcal{N}(X_i|\mu_0, \sigma^2) + 0.5\mathcal{N}(X_i|\mu_1, \sigma^2)] \cdot p(\theta)$$

The derivation of posterior of each parameter is not possible, so we introduce a latent variable Z_i. Here,

$$Z_i|\pi \sim \text{Bernoulli}(0.5)$$

Then, the posterior distribution is

$$p(\theta|\mathbf{X}, \mathbf{Z}) = \prod_{i=1}^{n} 0.5\mathcal{N}(X_i|\theta_{Z_i}) \cdot p(\theta)$$

The Gibbs sampling algorithm is:

- Initialization. Choose $\mu_0^{(0)}$, $\mu_1^{(0)}$, and $\sigma^{2(0)}$.
- Step t. For $t = 1, \ldots$
  - Generate $Z_i^{(t)}$ $(i = 1, \ldots, n)$ from

$$\mathbb{P}(Z_i^{(t)} = 0) = 1 - \mathbb{P}(Z_i^{(t)} = 1) = \frac{\mathcal{N}(X_i|\mu_0^{(t-1)}, \sigma^{2(t-1)})}{\mathcal{N}(X_i|\mu_0^{(t-1)}, \sigma^{2(t-1)}) + \mathcal{N}(X_i|\mu_1^{(t-1)}, \sigma^{2(t-1)})}$$

  - Compute $n_0^{(t)} = \sum_{i=1}^{n} \mathbb{I}(Z_i^{(t)} = 0)$ and $n_1^{(t)} = \sum_{i=1}^{n} \mathbb{I}(Z_i^{(t)} = 1)$.
  - Generate $\mu_0^{(t)}$ and $\mu_1^{(t)}$

$$\mu_0^{(t)}|\cdot \sim \mathcal{N}\left(\frac{\lambda m_0 + \sum_{i:Z_i=0} X_i}{\lambda + n_0^{(t)}}, \frac{1}{\lambda + n_0^{(t)}}\right)$$

$$\mu_1^{(t)}|\cdot \sim \mathcal{N}\left(\frac{\lambda m_1 + \sum_{i:Z_i=1} X_i}{\lambda + n_1^{(t)}}, \frac{1}{\lambda + n_1^{(t)}}\right)$$

  - Generate $\sigma^{2(t)}$

$$\sigma^{2(t)}|\cdot \sim \text{IG}\left(a + \frac{n}{2}, b + \frac{\sum_{i:Z_i=0}(X_i - \mu_0^{(t)})^2 + \sum_{i:Z_i=1}(X_i - \mu_1^{(t)})^2}{2}\right)$$

After implementing this algorithm 5000 times (see R script `exercise5_2`), we have the following graph (Figure 2) that shows the comparison of the true density (black) and the estimated Gaussian mixture density (purple). We can see that the estimated density fits very well. The means of $DinnerService = 0$ and $DinnerService = 1$ are -0.7619019 and 0.7558309 respectively, while the estimated means are -0.7601118 and 0.7584362, which are quite close to the sample mean.

We also plot the histogram of correct labeling ($d\_sample := DinnerService$) (Figure 3), from which we can see that the labeling is approximately half correct. This is due to `label switching` (i.e., the posterior distribution is invariant to switching component labels), which is a problematic issue when using MCMC to estimate mixture models.

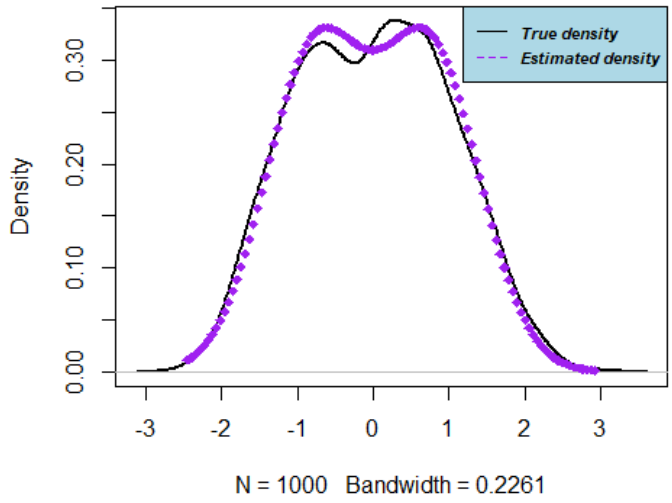


Figure 2: Comparison of true density and estimated density

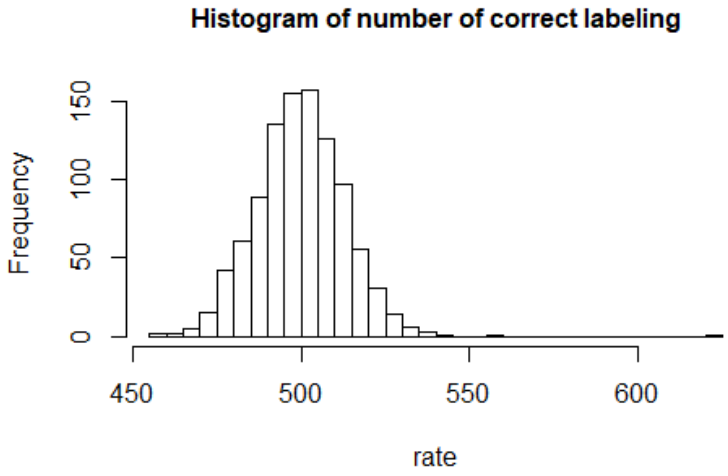


Figure 3: Histogram of number of correct labeling

OK, let's now assume we don't know π, and that the two classes have different values of σ^2. Let's put a Beta(α, β) prior on π, since it is conjugate to the Bernoulli distribution.

Exercise 5.3 *Assume integrating out π. What is the conditional distribution $P(Z_i|Z_{\neg i}, X_i, \mu_0, \mu_1, \sigma_0, \sigma_1, \alpha, \beta)$, where $Z_{\neg i}$ means all the values of Z except Z_i?*

Solution: Denote $\mathbf{X} = (X_1, \ldots, X_n)$, $\theta = (\mu_0, \mu_1, \tau_0, \tau_1)$, where $\tau_0 = 1/\sigma_0^2$, $\tau_1 = 1/\sigma_1^2$, and $\mathbf{Z} = (Z_1, \ldots, Z_n)$, then the model is

$$f(X_i|\pi, \theta) = \pi \mathcal{N}(X_i|\mu_0, \tau_0) + (1 - \pi)\mathcal{N}(X_i|\mu_1, \tau_1)$$

The prior of the parameters are

$$\begin{aligned}\mu_0 &\sim \mathcal{N}(m_0, 1/\lambda)\\ \mu_1 &\sim \mathcal{N}(m_1, 1/\lambda)\\ \tau_0 &\sim \text{Gamma}(a_0, b_0)\\ \tau_1 &\sim \text{Gamma}(a_1, b_1)\\ \pi &\sim \text{Beta}(\alpha, \beta)\end{aligned}$$

The the full likelihood is

$$f(\mathbf{X}|\pi, \theta) = \prod_{i=1}^{n} [\pi \mathcal{N}(X_i|\mu_0, \tau_0) + (1-\pi)\mathcal{N}(X_i|\mu_1, \tau_1)]$$

The posterior distribution of (π, θ) is

$$p(\pi, \theta|\mathbf{X}) = f(\mathbf{X}|\pi, \theta)p(\pi, \theta)$$

The derivation of posterior of each parameter is not possible, so we introduce a latent variable Z_i. Here,

$$Z_i|\pi \sim \text{Bernoulli}(\pi)$$

Then, our model becomes

$$f(X_i, Z_i|\theta) = \pi_{Z_i}\mathcal{N}(X_i|\theta_{Z_i})$$

The conditional distribution of Z_i is

$$p(Z_i = 1|Z_{\neg i}, X_i, \pi, \mu_0, \mu_1, \tau_0, \tau_1) = \frac{f(X_i|Z_i = 1, \theta)p(Z_i = 1)}{f(X_i|\pi, \theta)} = \frac{(1-\pi)\mathcal{N}(X_i|\mu_1, \tau_1)}{\pi\mathcal{N}(X_i|\mu_0, \tau_0) + (1-\pi)\mathcal{N}(X_i|\mu_1, \tau_1)}$$

$$p(Z_i = 0|Z_{\neg i}, X_i, \pi, \mu_0, \mu_1, \tau_0, \tau_1) = \frac{f(X_i|Z_i = 0, \theta)p(Z_i = 0)}{f(X_i|\pi, \theta)} = \frac{\pi\mathcal{N}(X_i|\mu_0, \tau_0)}{\pi\mathcal{N}(X_i|\mu_0, \tau_0) + (1-\pi)\mathcal{N}(X_i|\mu_1, \tau_1)}$$

Hence,

$$Z_i|Z_{\neg i}, X_i, \pi, \mu_0, \mu_1, \tau_0, \tau_1 \sim \text{Bernoulli}\left(\frac{(1-\pi)\mathcal{N}(X_i|\mu_1, \tau_1)}{\pi\mathcal{N}(X_i|\mu_0, \tau_0) + (1-\pi)\mathcal{N}(X_i|\mu_1, \tau_1)}\right)$$

But we need to integrate out π, so we turn to the completed likelihood is

$$f(\mathbf{X}, \mathbf{Z}|\pi, \theta) = L(\pi, \theta|\mathbf{X}, \mathbf{Z}) = \prod_{i=1}^{n} \pi_{Z_i}\mathcal{N}(X_i|\theta_{Z_i})$$

Let $n_k = \sum_{i=1}^{n} \mathbb{I}(Z_i = k)$, where $k \in \{0, 1\}$. Then, the joint distribution is

$$\begin{aligned}f(\mathbf{X}, \mathbf{Z}, \pi, \theta) &= f(\mathbf{X}, \mathbf{Z}|\pi, \theta)p(\pi, \theta)\\ &= \prod_{i=1}^{n} \pi_{Z_i}\mathcal{N}(X_i|\theta_{Z_i}) \times p(\pi, \theta)\\ &= f(\pi)f(\mu_0)f(\mu_1)f(\tau_0)f(\tau_1)\prod_{i=1}^{n} \pi_{Z_i}\mathcal{N}(X_i|\theta_{Z_i})\\ &= \frac{\pi^{\alpha+n_0-1}(1-\pi)^{\beta+n_1-1}}{B(\alpha+\beta)} f(\mu_0)f(\mu_1)f(\tau_0)f(\tau_1)\prod_{i=1}^{n} \mathcal{N}(X_i|\theta_{Z_i})\end{aligned}$$

By integrating out π, we have

$$f(\mathbf{X}, \mathbf{Z}, \theta) = \frac{B(\alpha+n_0, \beta+n_1)}{B(\alpha+\beta)} f(\mu_0)f(\mu_1)f(\tau_0)f(\tau_1)\prod_{i=1}^{n} \mathcal{N}(X_i|\theta_{Z_i})$$

$$\Rightarrow p_1 = p(Z_i = 1|Z_{\neg i}, X_i, \mu_0, \mu_1, \tau_0, \tau_1) \propto \frac{B(\alpha + n_0, \beta + n_1)}{B(\alpha + \beta)} \mathcal{N}(X_i|\mu_1, \tau_1)$$

$$\Rightarrow p_0 = p(Z_i = 0|Z_{\neg i}, X_i, \mu_0, \mu_1, \tau_0, \tau_1) \propto \frac{B(\alpha + n_0, \beta + n_1)}{B(\alpha + \beta)} \mathcal{N}(X_i|\mu_0, \tau_0)$$

which is a Bernoulli distribution with parameter $\frac{p_1}{p_0+p_1}$.

Exercise 5.4 *How about if we want to integrate out all of the continuous variables? What is the conditional distribution $P(Z_i|Z_{\neg i}, X, \theta)$, where θ is the set of all hyperparameters?*

Exercise 5.5 *Implement a Gibbs sampler for this new model where we learn the cluster proportions. You can either implement one of the variants in the previous two exercises, or the fully uncollapsed model where we sample Z, π, μ_0, μ_1, σ_0^2 and σ_1^2.*

Solution: From `exercise 5.4`, we can derive the posterior of π,

$$p(\pi|\mathbf{Z}, \mathbf{X}, \mu_0, \mu_1, \tau_0, \tau_1, \alpha, \beta) \propto \prod_{i=1}^{n} \pi_{Z_i} \mathcal{N}(X_i|\theta_{Z_i}) \times \text{Beta}(\alpha, \beta) = \text{Beta}(\alpha + n_0, \beta + n_1)$$

And conditional distribution of Z_i

$$Z_i|Z_{\neg i}, X_i, \pi, \mu_0, \mu_1, \tau_0, \tau_1 \sim \text{Bernoulli}\left(\frac{(1-\pi)\mathcal{N}(X_i|\mu_1, \tau_1)}{\pi\mathcal{N}(X_i|\mu_0, \tau_0) + (1-\pi)\mathcal{N}(X_i|\mu_1, \tau_1)}\right)$$

The uncollapsed Gibbs sampling algorithm is:

- Initialization. Choose $\mu_0^{(0)}$, $\mu_1^{(0)}$, $\tau_0^{(0)}$, $\tau_1^{(0)}$ and $\pi^{(0)}$.
- Step t. For $t = 1, \ldots$
  - Generate $\pi^{(t)} = \text{Beta}(1+n_0^{(t-1)}, 1+n_1^{(t-1)})$, where $n_0^{(t-1)} = \sum_{i=1}^{n} \mathbb{I}_{Z_i^{(t-1)}=0}$ and $n_1^{(t-1)} = \sum_{i=1}^{n} \mathbb{I}_{Z_i^{(t-1)}=1}$.
  - Generate $Z_i^{(t)}$ $(i = 1, \ldots, n)$ from

$$\mathbb{P}(Z_i^{(t)} = 0) = 1 - \mathbb{P}(Z_i^{(t)} = 1) = \frac{\pi^{(t)}\mathcal{N}(X_i|\mu_0^{(t)}, \tau_0^{(t)})}{\pi^{(t)}\mathcal{N}(X_i|\mu_0^{(t)}, \tau_0^{(t)}) + (1-\pi^{(t)})\mathcal{N}(X_i|\mu_1^{(t)}, \tau_1^{(t)})}$$

  - Generate $\mu_0^{(t)}$ and $\mu_1^{(t)}$

$$\mu_0^{(t)}|\cdot \sim \mathcal{N}\left(\frac{\lambda m_0 + \tau_0^{(t-1)} \sum_{i:Z_i^{(t)}=0} X_i}{\lambda + n_0^{(t)}\tau_0^{(t-1)}}, \frac{1}{\lambda + n_0^{(t)}\tau_0^{(t-1)}}\right)$$

$$\mu_1^{(t)}|\cdot \sim \mathcal{N}\left(\frac{\lambda m_1 + \tau_1^{(t-1)} \sum_{i:Z_i^{(t)}=1} X_i}{\lambda + n_1^{(t)}\tau_1^{(t-1)}}, \frac{1}{\lambda + n_1^{(t)}\tau_1^{(t-1)}}\right)$$

  - Generate $\tau_0^{(t)}$ and $\tau_1^{(t)}$

$$\tau_0^{(t)}|\cdot \sim \text{Gamma}\left(a_0 + \frac{n_0}{2}, b_0 + \frac{\sum_{i:Z_i^{(t)}=0}(X_i - \mu_0^{(t)})^2}{2}\right)$$

$$\tau_1^{(t)}|\cdot \sim \text{Gamma}\left(a_1 + \frac{n_1}{2}, b_1 + \frac{\sum_{i:Z_i^{(t)}=1}(X_i - \mu_1^{(t)})^2}{2}\right)$$

In our implementation, we will set same variance of the two clusters to guarantee the identifiability of our model. For the prior, we set $a = b = 1$, $m_0 = 1$, $m_1 = -1$, and $\lambda = 1$. After implementing the algorithm 10000 times and using burnin = 2001, we have the following histograms of μ_0, μ_1, σ^2 and weight π (See Figure 4). The histogram of the weight shows that the estimated mean weight is equal 0.5018532, indicating that the proportions of the two clusters are roughly equal. The estimated mean of both μ_0 and μ_1 are -0.765392 and 0.758918, which are quite close to the means of our data. Figure 5 shows the comparison of the true density and the estimated density. We can see that the estimated density fits very well and the plot are quite similar to what we got in `exercise 5.2`.

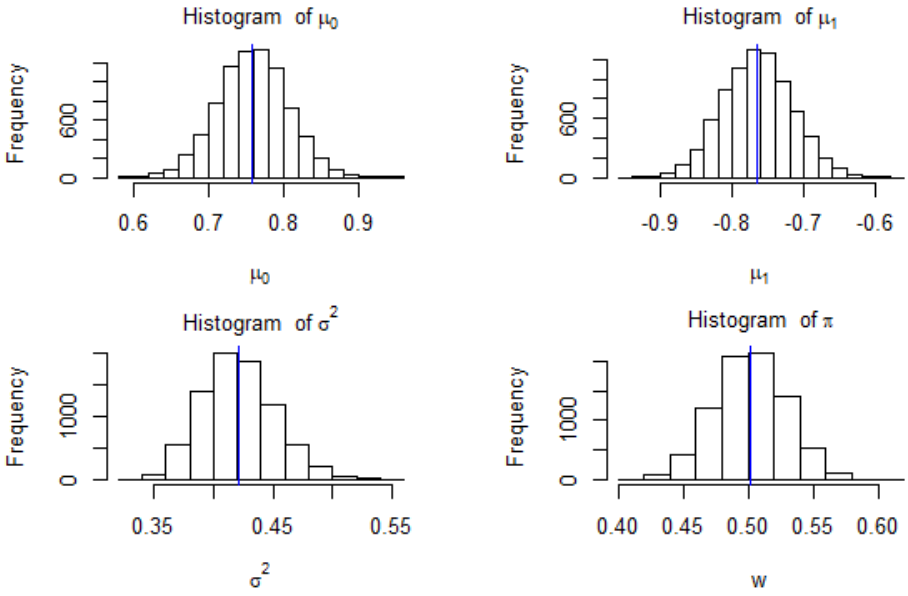


Figure 4: Histograms of μ_0, μ_1, σ^2 and weight π

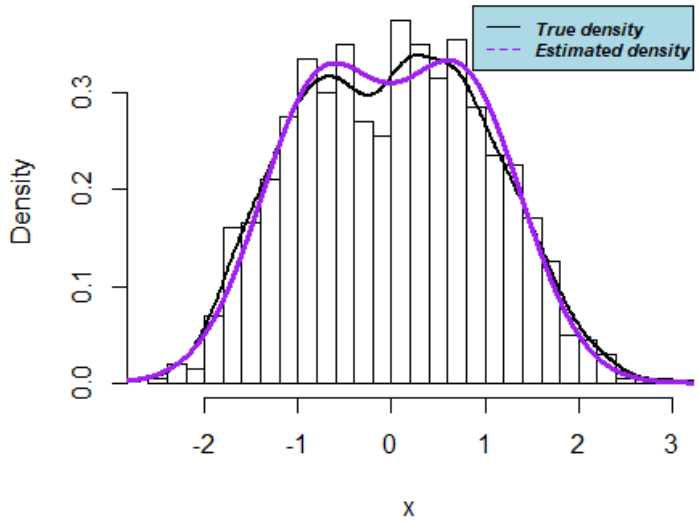


Figure 5: Comparison of true density and estimated density

Let's now consider the case where we have more than two classes. Here, we need to replace our Bernoulli distribution with a multinomial parametrized by some probability vector π, so that:

$$P(Z_i = k) = \pi_k$$

Exercise 5.6 *Much as the multinomial is the multivariate generalization of the binomial distribution, the Dirichlet($\alpha_1, \ldots, \alpha_K$) distribution, which has pdf*

$$\frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1},$$

is the multivariate generalization of the beta distribution. Show that the Dirichlet is conjugate to the multinomial, and derive the posterior predictive distribution

$$P(Z_{n+1}|Z_{1:n}) = \int_{\mathcal{M}} P(Z_{n+1}|\pi)p(\pi)d\pi$$

You may find it helpful to note that, if $\pi \sim$ Dirichlet($\alpha_1, \ldots, \alpha_K$), then $E[\pi] = \frac{(\alpha_1, \ldots, \alpha_K)}{\sum_k \alpha_k}$.

Exercise 5.7 *Modify your previous Gibbs sampler to allow multiple classes, and two-dimensional data. Generate some data according to a Dirichlet mixture of 5 Gaussians in $\mathbb{R}^2$, and test your code on it.*

Exercise 5.8 *OK, let's try a real dataset! We're going to use a set of images from MNIST. Download the dataset mnist.csv from the data directory, and transform it to be zero mean, unit variance. Each row contains the vectorized pixel values for an image of a digit. The whole dataset contains 100 copies of each digit, with the first 100 being zeros, the next 100 being ones, etc. You can visualize a data point by reshaping it to be 28×28:*

- *R:* `image(matrix(X[1,],nrow=28))`
- *Python:* `import matplotlib.pyplot; plt.imshow(X[0,:].reshape(28,28)); plt.show()`
- *Matlab:* `imshow(reshape(X(1,:),28,28))`

The data is 784-dimensional; let's reduce this by running PCA and using the first 50 dimensions.

Now, try running your Gibbs sampler with 10 classes, and $\alpha_1 = \alpha_2 = \cdots = \alpha_{10} = 1$. This prior corresponds to a uniform distribution on the 9-simplex. It's fine to use a spherical covariance here... in fact it will work fine if you just have a prior on the means, and fix $\sigma^2 = 1$.

Here are some ways you can visualize your output:

- *Based on a single sample, plot the recovered clustering vs the ground truth clustering.*
- *Based on a single sample, visualize the mean image for each cluster, by multiplying the mean embedding with the coefficients obtained using PCA.*
- *Over multiple samples, create a co-occurrence matrix with entries being the proportion of the times that the two data points are in the same sample.*

Exercise 5.9 *OK, let's try a different likelihood. Let's consider modeling documents. A common modeling assumption is to treat a document as a "bag-of-words" – assuming that all the information is in the words, and none of it is in the ordering. Under this assumption, an appropriate distribution is a multinomial distribution over words, with a Dirichlet prior. Concretely, let:*

$$\begin{aligned}
\pi \sim & Dirichlet_K(\alpha) \\
\eta_k \sim & Dirichlet_V(\beta), \qquad k = 1, \ldots, K \\
z_i \sim & Discrete(\pi), \qquad i = 1, \ldots, D \\
\mathbf{w}_i \sim & Multinomial(\eta_{z_i})
\end{aligned}$$

where D is the number of documents, V is the number of words in the dictionary, K is the number of clusters, and $\mathbf{w}_i$ is a V-dimensional count vector representing the ith document.

Write out the conditional distributions for a collapsed (i.e. integrating out π and the η_k) Gibbs sampler for this model.

Solution: Note that the notations in this problem will remain unchanged, but in `exercise 5.11` new notations corresponding to the plate notation will be added.

The generative process of words in a document are as following:

- For each of the K topic: choose $\eta_k \sim Dir_V(\beta)$
- For each document in the corpus: choose topic distribution $\pi \sim Dir_K(\alpha)$
- For each of the m_i word occurrences in the document i:
  - choose a topic $z_i \sim Discrete(\pi)$
  - choose a word from the corresponding topic $w_i \sim Mult(\eta_{z_i})$

Here presents a simple explanation of the notations:

- α is K (number of topics) dimensional hyperparameter for topic distribution.
- β is V (number of words) dimensional hyperparameter for word distribution over a given topic.
- η_{kw} is the probability of choosing word w given it is drawn from topic k.
- π is the topic distribution for each document.
- z_i is the corresponding topic for the each word in document i.

Later on, I am also going to use the following notations:

- $\mathbf{w} = \{w_i\}$, $w_i = \{w_{in}\}$, are the observed words.
- $\mathbf{z} = \{z_i\}$, $z_i = \{z_{in}\}$, are the topic assignments for word occurrence.
- n_k is a V dimensional vector, where n_{kw} is the count of word w in the corpus assigned to topic k.
- m_i is a K dimensional vector, where m_{ik} is the count of words in document i assigned for topic k.

Also, note that

(1) Dirichlet distribution:

$$Dir(\theta;\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{|\alpha|} \theta_i^{\alpha_i - 1}$$

(2) Multinomial Beta function:

$$B(\alpha) = \frac{\prod_{i=1}^{|\alpha|} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{|\alpha|} \alpha_i)}$$

(3) Gamma function

$$\Gamma(n) = (n-1)!$$

(4) Multinomial distribution:

$$Mult(x;\theta) = \frac{n!}{\prod_{i=1}^{|x|} x_i!} \prod_{i=1}^{|x|} \theta_i^{x_i}$$

(5) Posterior of a multinomial:

$$Dir(\theta; x+\alpha) = Mult(x;\theta) Dir(\theta;\alpha)$$

$$\Rightarrow \int Dir(\theta; x+\alpha) d\theta = \int \frac{1}{B(x+\alpha)} \prod_{i=1}^{|\alpha|} \theta^{x_i+\alpha_i-1} d\theta = 1 \qquad \Rightarrow \int \prod_{i=1}^{|\alpha|} \theta^{x_i+\alpha_i-1} d\theta = B(x+\alpha)$$

Then the conditional distribution is

$$\begin{aligned}
p(\mathbf{w}, \mathbf{z}|\alpha, \beta) &= \prod_i p(w_i, z_i|\alpha, \beta) \\
&= \prod_i p(w_i|z_i, \beta) \prod_i p(z_i|\alpha) \\
&= \int \prod_i p(w_i|z_i, \eta) p(\eta|\beta) d\eta \times \prod_i \int p(z_i|\pi) p(\pi|\alpha) d\pi \\
&= \underbrace{\int \prod_k \prod_w \eta^{n_{kw}} \cdot \prod_k \frac{1}{B(\beta)} \prod_w \eta^{\beta_w - 1} d\eta}_{} \times \underbrace{\prod_i \int \prod_k \pi^{m_{ik}} \cdot \frac{1}{B(\alpha)} \prod_k \pi^{\alpha_k - 1} d\pi}_{} \\
&= \underbrace{\frac{1}{(B(\beta))^K} \int \prod_k \prod_w \eta^{n_{kw} + \beta_w - 1} d\eta}_{} \times \underbrace{\prod_i \frac{1}{B(\alpha)} \int \prod_k \pi^{m_{ik} + \alpha_k - 1} d\pi}_{} \\
&= \prod_k \frac{B(n_k + \beta)}{B(\beta)} \prod_i \frac{B(m_i + \alpha)}{B(\alpha)}
\end{aligned}$$

Exercise 5.10 *Implement the code. Generate a test set by generating data from a mixture of two multinomials, one with probabilities* $(1,1,1,1,9,9,9,9)/40$ *and the other with probabilities* $(9,9,9,9,1,1,1,1)/40$. *Test your code on this dataset, and compare a single sample's clustering pattern with the ground truth values.*

Once you've got it to work on the toy data, try it on some real data! The file `cora.csv` *on Github contains a bag-of-words representation of a collection of 2410 scientific documents from the Cora search engine (taken*

from the R package `lda`*. Each row corresponds to a document, each column to a word, each element is the number of times that word appears in that document. The list of words is at* `cora_vocab.csv`*. Try clustering them into say 10 clusters. The NIPS dataset on Github contains the text of NIPS papers. Try clustering them into say 10 clusters. Based on a single sample for each cluster, report the 10 most frequently occurring words.*

5.0.1 Admixture models

A mixture model for text isn't massively realistic. Consider the NIPS papers: is it really reasonable to separate multiple documents into distinct clusters? It is more likely that two papers share some aspects in common, but differ on others.

We can use a hierarchical Bayesian formulation to model each document using a mixture model, with a shared prior on the mixing components. Concretely, let

$$\begin{aligned}
\theta_d &\sim \text{Dirichlet}_K(\alpha), \qquad d = 1, \ldots, D \\
\beta_k &\sim \text{Dirichlet}_V(\eta), \qquad k = 1, \ldots, K \\
z_{dn} &\sim \text{Multinomial}(\theta_i), \qquad j = 1, \ldots, N_d \\
w_{dn} &\sim \text{Multinomial}(\beta_{z_{dn}}),
\end{aligned}$$

where N_d is the number of words in the dth document. This model is commonly known as Latent Dirichlet Allocation (see `http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf`); it is an example of an *admixture* model. This means that each document is associated with a distribution θ_d over clusters, and each word is associated with a single cluster.

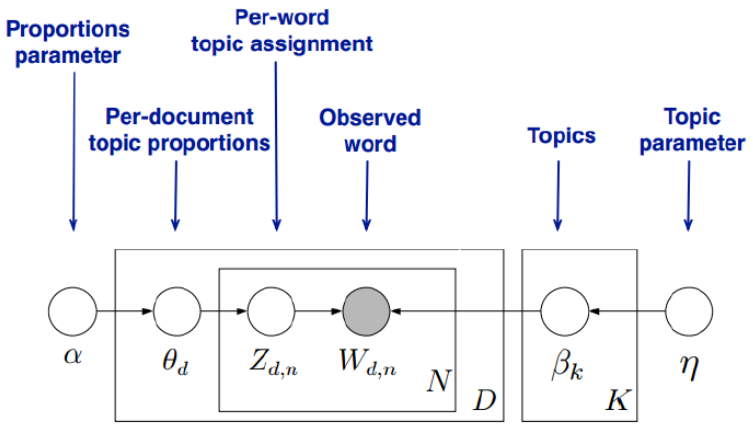
Exercise 5.11 *We can construct a collapsed Gibbs sampler for this model by integrating out the θ_d and the β_k. Derive the predictive distributions $p(z_{dn}|\{z_{\neg dn}\}, \alpha)$ and $p(w_{dn}|z_{dn}, z_{\neg dn}, w_{\neg dn}, \beta)$, and hence the conditional distribution $p(z_{dn}|rest)$*

Solution: Latent Dirichlet Allocation is a generative graphical model for mining latent topics of texts or any data with similar underlying statistical structures.

The generative process of words in a document are as following:

- For each of the K topic: choose $\beta_k \sim Dir(\eta)$
- For each document d in the corpus: choose topic distribution $\theta_d \sim Dir(\alpha)$
- For each of the N_d word occurrences in the document d:
  - choose a topic $z_{dn} \sim Mult(\theta_d)$
  - choose a word from the corresponding topic $w_{dn} \sim Mult(\beta_{z_{dn}})$

Below is the plate notation representing the LDA model.



Here presents a simple explanation of the notations:

- α is K (number of topics) dimensional hyperparameter for topic distribution.
- η is V (number of words) dimensional hyperparameter for word distribution over a given topic.
- β_{kw} is the probability of choosing word w given it is drawn from topic k.
- θ_d is the topic distribution for document d.
- z_{dn} is the corresponding topic for the n-th word in document d.

Later on, I am also going to use the following notations:

- $\mathbf{w} = \{w_d\}$, $w_d = \{w_{dn}\}$, are the observed words.
- $\mathbf{z} = \{z_d\}$, $z_d = \{z_{dn}\}$, are the topic assignments for word occurrence.
- n_k is a $|V|$ dimensional vector, where n_{kw} is the count of word w in the corpus assigned to topic k.
- m_d is a K dimensional vector, where m_{dk} is the count of words in document d assigned for topic k.

What we are interested in this model are random variables β, θ, and maybe z. But before going to inference of these latent random variables, we may first follow the routine and write down the likelihood of the data:

$$p(\beta, \theta, \mathbf{z}, \mathbf{w}) = \left(\prod_{k=1}^{K} p(\beta_k|\eta)\right)\left(\prod_{d=1}^{D} p(\theta_d|\alpha)\prod_{n=1}^{N_d} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn}, \beta)\right)$$

where

$$p(\beta|\eta) = \prod_{k}^{K} \frac{\Gamma(\sum_{v=1}^{V}\eta_v)}{\prod_{v=1}^{V}\Gamma(\eta_v)}\beta_{k1}^{\eta_1-1}\cdots\beta_{kV}^{\eta_V-1}$$

$$p(\theta_d|\alpha) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\theta_{d1}^{\alpha_1-1}\cdots\theta_{dK}^{\alpha_K-1}$$

$$p(z_{dn}|\theta_d) = \theta_{d,z_{dn}}$$

$$p(w_{dn}|z_{dn},\beta) = \beta_{z_{dn},w_{dn}}$$

Then we derive the conditional joint distribution as we did in `exercise 5.9`,

$$\begin{aligned}
p(\mathbf{w},\mathbf{z}|\alpha,\eta) &= \prod_d p(w_d,z_d|\alpha,\eta) \\
&= \prod_d p(w_d|z_d,\eta)\prod_d p(z_d|\alpha) \\
&= \int \prod_d p(w_d|z_d,\beta)p(\beta|\eta)d\beta \times \prod_d \int p(z_d|\theta_d)p(\theta_d|\alpha)d\theta_d \\
&= \underbrace{\int \prod_k \prod_w \beta^{n_{kw}} \cdot \prod_k \frac{1}{B(\eta)} \prod_w \beta^{\eta_w - 1} d\beta}_{} \times \underbrace{\prod_d \int \prod_k \theta_d^{m_{dk}} \cdot \frac{1}{B(\alpha)} \prod_k \theta_d^{\alpha_k - 1} d\theta_d}_{} \\
&= \underbrace{\frac{1}{(B(\eta))^K} \int \prod_k \prod_w \beta^{n_{kw}+\eta_w-1} d\beta}_{} \times \underbrace{\prod_d \frac{1}{B(\alpha)} \int \prod_k \theta_d^{m_{dk}+\alpha_k-1} d\theta_d}_{} \\
&= \prod_k \frac{B(n_k+\eta)}{B(\eta)} \prod_d \frac{B(m_d+\alpha)}{B(\alpha)}
\end{aligned}$$

Now we can start to derive key formula for Gibbs sampling, which is $p(z_i|z_{\neg i},\mathbf{w},\alpha,\eta)$, here $z_{\neg i}$ means the topic assignments for all word occurrences other than the one at position i, without loss of generality, we assume here that the position i is in document d and associated with the observed word w, since

$$p(z_i|z_{\neg i},\mathbf{w},\alpha,\eta) = \frac{P(\mathbf{z},\mathbf{w}|\alpha,\eta)}{p(z_{\neg i},\mathbf{w}|\alpha,\eta)} \propto \frac{p(\mathbf{z},\mathbf{w}|\alpha,\eta)}{p(z_{\neg i},w_{\neg i}|\alpha,\eta)}$$

Then, we have

$$\begin{aligned}
p(z_i = k|_{\neg i},\mathbf{w},\alpha,\eta) &\propto \frac{p(z_i=k,z_{\neg i},\mathbf{w}|\alpha,\eta)}{p(z_{\neg i},w_{\neg i}|\alpha,\eta)} \\
&= \frac{B(n_k+\eta)}{B(n_k^{\neg i}+\eta)} \frac{B(m_d+\alpha)}{B(m_d^{\neg i}+\alpha)} \\
&= \frac{n_{kw}+\eta_w-1}{[\sum_{w'} n_{kw'}+\eta_{w'}]-1} \frac{m_{dk}+\alpha_k-1}{[\sum_{k'} m_{dk'}+\alpha_{k'}]-1} \\
&\propto \frac{n_{kw}^{\neg i}+\eta_w}{\sum_{w'} n_{kw'}^{\neg i}+\eta_{w'}} (m_{dk}^{\neg i}+\alpha_k)
\end{aligned}$$

Exercise 5.12 *I'm not going to make you implement this one (although if you want to, feel free!). Instead, let's use the R package* `lda` *(sorry Python/R folk! it should be fairly easy to use). The documentation is here:* `https://cran.r-project.org/web/packages/lda/lda.pdf`*. Run the Gibbs sampler on the built-in document dataset* `cora`*, and report the 5 words with highest probability for each cluster (hint: look at the example under top.topic.words – note that you might need more iterations than is given in the example, R has a rule that examples have to run quickly, hence the low number in the example). Why is this sort of model commonly called a topic model?*

5.1 Bayesian nonparametric models

When we were modeling the MNIST dataset, we used 10 clusters. This seems reasonable, right – there are 10 digits! However, if you look at the data, there is a lot of variation within each digit. Maybe we'd be better off using more clusters... but how many?

One answer to this question is to allow *infinitely* many clusters *a priori*. Each data point can only belong to a single cluster, so there will only be at most N occupied clusters. By allowing infinitely many clusters, we can allow N data points to occupy a random number of clusters. Further, if we see more data, we are not restricted to the previously occupied clusters.

Exercise 5.13 *To get a feel for this, we can "approximate" a model with infinitely many clusters with a model with a large number of clusters. Let's start with a Dirichlet prior on cluster membership, with 100 clusters. Sample* $\pi \sim Dirichlet_{100}(10, 10, \ldots, 10)$*, and then sample 10 cluster indicators* $z_i \sim \pi$*. Record the list of cluster indicators, e.g.* $\{1, 10, 11, 11, \ldots\}$*. Do this 5 times, with a different* π *each time.*

Repeat this with $\alpha = (1, 1, \ldots, 1)$*,* $\alpha = (0.1, 0.1, \ldots, 0.1)$ *and* $\alpha = (0.01, 0.01, \ldots, 0.01)$*.*

Comment on how the value of α *affects your clustering behavior.*

OK, now let's explore some further properties of the Dirichlet distribution. First, we note an important relationship between the Dirichlet distirbution from the gamma distribution: If

$$\gamma_i \overset{\text{iid}}{\sim} \text{Gamma}(\alpha_i, \beta)$$

then

$$Z = \sum_{i=1}^{K} \gamma_i \sim \text{Gamma}\left(\sum_{i=1}^{K} \alpha_i, \beta\right)$$

and

$$\pi = \left(\frac{\gamma_1}{Z}, \ldots, \frac{\gamma_K}{Z}\right) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$$

Exercise 5.14 *Using the change-of-variable technique with the transform* $(\gamma_1, \ldots, \gamma_K) \to (\pi_1, \ldots, \pi_{K-1}, Z)$*, prove the above result.*

Solution: The transformation $(\gamma_1, \ldots, \gamma_K) \mapsto (\pi_1 \ldots, \pi_{K-1}, Z)$ is one-to-one: its inverse is given by

$$\gamma_k = \pi_k Z \;\text{ for }\; 1 \le k \le K-1 \;\text{ and }\; \pi_K = z(1 - \pi_1 - \cdots - \pi_{K-1})$$

Then the new probability element simply is

$$\begin{aligned}&(z\pi_1)^{\alpha_1 - 1} \ldots (z\pi_{K-1})^{\alpha_{K-1}-1}(z(1 - \pi_1 - \cdots - \pi_{K-1}))^{\alpha_K - 1} e^{-\beta z} |z^{K-1} d\pi_1 \wedge \cdots \wedge d\pi_{K-1} dz| \\ =&(z^{\alpha_1 + \cdots + \alpha_K - 1} e^{-\beta z} dz)(\pi_1^{\alpha_1 - 1} \ldots \pi_{K-1}^{\alpha_{K-1}-1} d\pi_1 \ldots d\pi_{K-1})\end{aligned}$$

which is a product of a $\text{Gamma}(\alpha_1 + \cdots + \alpha_K, \beta)$ distribution for Z and a Dirichlet α distribution for $(\pi_1, \ldots, \pi_{K-1})$. Since the original normalizing constant must have been a product of $\Gamma(\alpha_k)$, we deduce immediately that the new normalizing constant must be divided by $\Gamma(\alpha_1, \ldots, \alpha_K)$, enabling the PDF to be written

$$f_\pi(\pi, \alpha) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_K)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \left(\pi_1^{\alpha_1 - 1} \cdots \pi_{K-1}^{\alpha_{K-1}-1} (1 - \pi_1 - \cdots - \pi_{K-1})^{\alpha_K - 1}\right)$$

Therefore, $\pi = \left(\frac{\gamma_1}{Z}, \ldots, \frac{\gamma_K}{Z}\right) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$.

You will probably find this relationship helpful in proving the following.

Exercise 5.15 (Agglomeration property) *Show that, if* $(\pi_1, \ldots, \pi_K) \sim$ *Dirichlet*$(\alpha_1, \ldots, \alpha_K)$*, then* $(\pi_1 + \pi_2, \pi_3, \ldots, \pi_K) \sim$ *Dirichlet*$(\alpha_1 + \alpha_2, \alpha_3, \ldots, \alpha_K)$.

Proof: Suppose $y_k \sim \text{Gamma}(\alpha_k, 1)$, let $z = \sum_{k=1}^K y_k$, $\pi_k = \frac{y_k}{z}$. Then from `exercise 5.14`, we have

$$(\frac{y_1}{z}, \ldots, \frac{y_K}{z}) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$$

On the other hand,

$$y_1 + y_2 \sim \text{Gamma}(\alpha_1 + \alpha_2, 1)$$

Hence, we have

$$(\pi_1 + \pi_2, \pi_3, \ldots, \pi_K) = (\frac{y_1 + y_2}{z}, \frac{y_3}{z}, \ldots, \frac{y_K}{z}) \sim \text{Dirichlet}(\alpha_1 + \alpha_2, \alpha_3, \ldots, \alpha_K)$$

■

Exercise 5.16 *Let* $\pi \sim \text{Dirichlet}_K\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right)$*, and assign weight* π_k *to the interval* $\left[\frac{k-1}{K}, \frac{k}{K}\right)$*. Show that, for any partition with breaks at multiples of* $\frac{1}{k}$*, the distribution over the weights associated with the blocks in the partition will be Dirichlet distributed.*

Proof: Suppose $y_k \sim \text{Gamma}(\alpha_k, 1)$, let $z = \sum_{k=1}^K y_k$, $\pi_k = \frac{y_k}{z}$. Then from `exercise 5.14`, we have

$$(\frac{y_1}{z}, \ldots, \frac{y_K}{z}) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$$

Now let $(I_1, \ldots, I_s)$ be a partition of $\{1, \ldots, K\}$ such that the interval $\left[\frac{i}{K}, \frac{j}{K}\right)$ of certain subset $I_{\{\cdot\}}$ is at multiples of $\frac{1}{k}$, where integers i, j take values from 1 to K. Now,

$$(\pi_1, \ldots, \pi_s) = \left(\frac{\sum_{k \in I_1} y_k}{z}, \ldots, \frac{\sum_{k \in I_s} y_k}{z}\right)$$

$$\sum_{k \in I_{\{\cdot\}}} y_k \sim \text{Gamma}\left(\sum_{k \in I_{\{\cdot\}}} \alpha_k, 1\right)$$

Therefore, we have

$$(\pi_1, \ldots, \pi_s) = \left(\frac{\sum_{k \in I_1} y_k}{z}, \ldots, \frac{\sum_{k \in I_s} y_k}{z}\right) \sim \text{Dirichlet}\left(\sum_{k \in I_1} \alpha_k, \ldots, \sum_{k \in I_s} \alpha_k\right)$$

which means that for any partition with breaks at multiples of $\frac{1}{k}$, the distribution over the weights associated with the blocks in the partition will be Dirichlet distributed. ■

The Dirichlet process extends this idea to arbitrary partitions. Concretely, the Dirichlet process is a distribution over measures[1] on some space $\otimes$, parametrized by some probability distribution H on Ω and some positive scalar α such that for any partition $A_1, \ldots, A_K$ of Ω, the masses assigned to $A_1, \ldots, A_k$ are distributed according to a Dirichlet $(\alpha H(A_1), \ldots, \alpha H(A_K))$ distribution. The resulting probability distribution D will have its probability concentrated on infinitely many singletons $D = \sum_{i=1}^\infty \pi_i \delta_{\theta_i}$– what is known as an atomic probability distribution.

[1] If you're not familiar with measure theory, a measure on some space is just a function that assigns a positive number to every subset of that space. So, a probability is a measure. Area is a measure.

We can construct a finite dimensional approximation to the Dirichlet process by sampling

$$\pi \sim \text{Dirichlet}_K\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right)$$

for some large α, and associating each probability π_k with a location $\theta_k \sim H$. This distribution will converge weakly to the Dirichlet process as $K \to \infty$.

Exercise 5.17 *Return to the MNIST mixture model, and replace your 10-dimensional Dirichlet distribution with a 100-dimensional Dirichlet with parameters $\alpha/100$ for, say, $\alpha = 1$. How many clusters does it use (look at a distribution over multiple samples)? Based on a single sample, what do those clusters look like?*