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:
The distribution of the profit has a heavy tail. It is not perfectly normal.

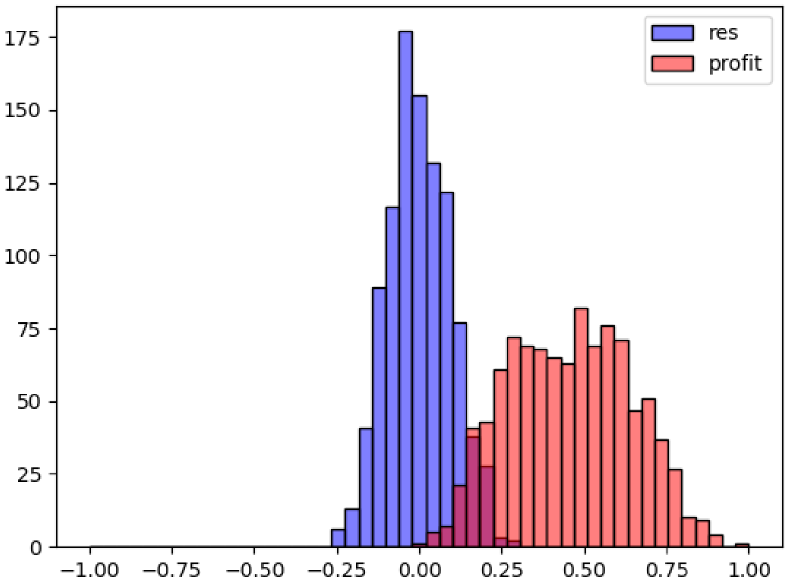

Figure 5.1: Histogram for the probability of belonging to class 3.

(*The code can be found on Github under python directory with name CH5_1.py*)

Solution End

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:

$$X_i \sim 0.5N(\mu_1, \sigma_1^2) + 0.5(\mu_2, \sigma_2^2)$$

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

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

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* $p(\mu_k|\{X_i : Z_i = k\}, \theta)$*, where* θ *are appropriate (shared) prior parameters for* μ_k*, implement a Gibbs sampler that samples the means and the latent indicator variables. I'd suggest using the parameters of the initial regression to pick your hyperparameters.*

Compare the clustering obtained with the "true" clustering due to the DinnerService variable.

Solution
The results of the clustering does not match exactly the expectation. Also, the classes can be flipped throughout the sampling process. The results show that 153 out of the 1000 samples clustered in the wrong group

(*The code can be found on Github under python directory with name CH5_2.py*)

solution End

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 *Let's assume we want to integrate out π. What is the conditional distribution $P(Z_i|Z_{\neg i}, X_i, \mu_1, \mu_2, \sigma_1, \sigma_2, \alpha,$ where $Z_{\neg i}$ means all the values of Z except Z_i?*

Solution:

$$\mathbf{P}(Z_i = z|X_i, X, Z_{\neg i}, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \alpha, \beta) \propto P(Z_i = z|\alpha, \beta, Z_{\neg i})p(X_i|X, \mu_z, \sigma_z^2)$$

The params for the posterior Bernoulli are:

$$\begin{aligned}
\alpha_n &= \alpha + \sum Z_{\neg i} \\
\beta_n &= \beta + n - \sum Z_{\neg i} \\
P(Z_i = 1|Z_{\neg i}) &= \frac{\alpha_n}{\alpha_n + \beta_n} \\
P(Z_i = 0|Z_{\neg i}) &= \frac{\beta_n}{\alpha_n + \beta_n}
\end{aligned}$$

The params for the posterior Gaussian are:

$$\begin{aligned}
\mu_{zn} &= \sigma_{zn}^2(\frac{\mu_z}{\sigma_0^2} + \frac{\sum_{Z_{\neg i,j}=1} Xj}{\sigma_z^2}) \\
\sigma_{zn}^2 &= (\frac{n_z}{\sigma_z^2} + \frac{1}{\sigma_0^2})^{-1} \\
p(X_i|X, \mu_z, \sigma_z^2) &\sim \text{Normal}(\mu_{zn}, \sigma_0^2 + \sigma_{zn}^2)
\end{aligned}$$

So,

$$\begin{aligned}
\mathbf{P}(Z_i = 1|X_i, X, Z_{\neg i}, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \alpha, \beta) &\propto \frac{\alpha_n}{\alpha_n + \beta_n}\text{Normal}(\mu_{1n}, \sigma_0^2 + \sigma_{1n}^2) \\
\mathbf{P}(Z_i = 0|X_i, X, Z_{\neg i}, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \alpha, \beta) &\propto \frac{\beta_n}{\alpha_n + \beta_n}\text{Normal}(\mu_{0n}, \sigma_0^2 + \sigma_{0n}^2)
\end{aligned}$$

Solution End

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?*

Solution: Assuming Normal inverse gamma prior $(\mu, \nu, \alpha, \beta)$, based on results of Section 2, the conditional distribution is given by a t-distribution with the following params:

$$\begin{aligned}
\mathbf{P}(Z_i = 1|Z_{\neg i}, X, \theta) &\propto \frac{\alpha_n}{\alpha_n + \beta_n}\text{t-dist}(\mu_{1n}, \frac{\beta_{1n}(\nu_{1n} + 1)}{\alpha_{1n}\nu_{1n}}) \\
n_1 &= \sum_{Z_{\neg i}} Z_j \\
\mu_{1n} &= \frac{\nu\mu + \sum_{Z_{\neg i}} X_j}{\nu + n_1} \\
\nu_{1n} &= \nu + n_1 \\
\alpha_{1n} &= \alpha + \frac{n_1}{2} \\
\beta_{1n} &= \beta + \frac{1}{2}\left\{\sum_{Z_{\neg i}} X_j^2 + \frac{\nu n_1}{\nu + n_1}\frac{(n_1^{-1}\sum_{Z_{\neg i}} X_j - \mu)^2}{2}\right\}
\end{aligned}$$

$\mathbf{P}(Z_i = 0|Z_{\neg i}, X, \theta)$ can be obtained in a similar way

Solution End

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_1, μ_2, σ_1^2 and σ_2^2.*

Solution:

The Gibbs sampler is implemented such that in each iteration the mean and the sigma (here sigma assumed constant) can be sampled. Then Z is sampled. The number of mismatches is still 153 like in the previous case.
(*The code can be found on Github under python directory with name CH5_5.py*)

Solution End

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*

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

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 $E[\pi] = \frac{(\alpha_1,\ldots,\alpha_K)}{\sum_k \alpha_k}$.

Solution:
Start from:

- $P(Z_i|\pi_1, \ldots, \pi_k) = \prod_{j=1}^{K} \pi_j^{z_i^{(j)}}$
- $P(\pi|\alpha) \propto \prod_{j=1}^{K} \pi_j^{\alpha_j}$

$$P(\pi|\alpha, Z) \propto \prod_{j=1}^{K} \pi_j^{\alpha_j} \prod_{i=1}^{n} \prod_{j=1}^{K} \pi_j^{1\{z_i=j\}} = \prod_{j=1}^{K} \pi_j^{\alpha_j + \sum 1\{z_i=j\}}$$

which is a new dirichlit distribution with updated parameters:

$$\alpha_j^n = \alpha_j + \sum 1\{z_i = j\}$$

The predictive distribution is:

$$\begin{aligned} P(Z_{n+1}|Z_{1:n}) = \int_{\mathcal{M}} P(Z_{n+1}|\pi)p(\pi)d\pi &= \int \frac{\Gamma(n+1)}{\prod_{j=1}^K \Gamma(z_{n+1}^{(j)}+1)} \prod_{j=1}^K \pi_j^{z_{n+1}^{(j)}} \frac{\Gamma(\sum_{j=1}^K \alpha_j^n)}{\prod_{j=1}^K \Gamma(\alpha_j^n)} \prod_{j=1}^K \pi_j^{\alpha_j^n} d\pi \\ &= \frac{\Gamma(n+1)}{\prod_{j=1}^K \Gamma(z_{n+1}^{(j)}+1)} \frac{\Gamma(\sum_{j=1}^K \alpha_j^n)}{\prod_{j=1}^K \Gamma(\alpha_j^n)} \int \prod_{j=1}^K \pi_j^{z_{n+1}^{(j)}} \prod_{j=1}^K \pi_j^{\alpha_j^n} d\pi \\ &= \frac{\Gamma(n+1)}{\prod_{j=1}^K \Gamma(z_{n+1}^{(j)}+1)} \frac{\Gamma(\sum_{j=1}^K \alpha_j^n)}{\prod_{j=1}^K \Gamma(\alpha_j^n)} \int \prod_{j=1}^K \pi_j^{\alpha_j^n + z_{n+1}^{(j)}} d\pi \\ &= \frac{\Gamma(n+1)}{\prod_{j=1}^K \Gamma(z_{n+1}^{(j)}+1)} \frac{\Gamma(\sum_{j=1}^K \alpha_j^n)}{\prod_{j=1}^K \Gamma(\alpha_j^n)} \frac{\Gamma(\sum_{j=1}^K z_{n+1}^{(j)} + \alpha_j^n)}{\prod_{j=1}^K \Gamma(\alpha_j^n + n)} \\ P(Z_{n+1} = k|Z_{1:n}) &= \frac{\alpha_k^n}{\sum_{j=1}^K \alpha_j^n} \end{aligned}$$

Solution End

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.*

Solution: This exercise is an preparing step for the next exercise. The next exercise is addressed directly. Solution of the next exercise include that of this one..

Solution End

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.*

Solution:
The figures below show the mean of each cluster for different runs of the sampler.

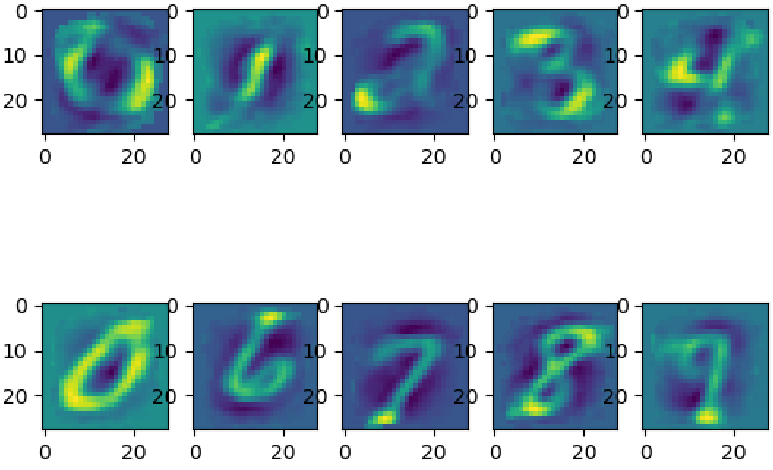

Figure 5.2: Running 700 iterations while initialized to the ‘golden solution’.

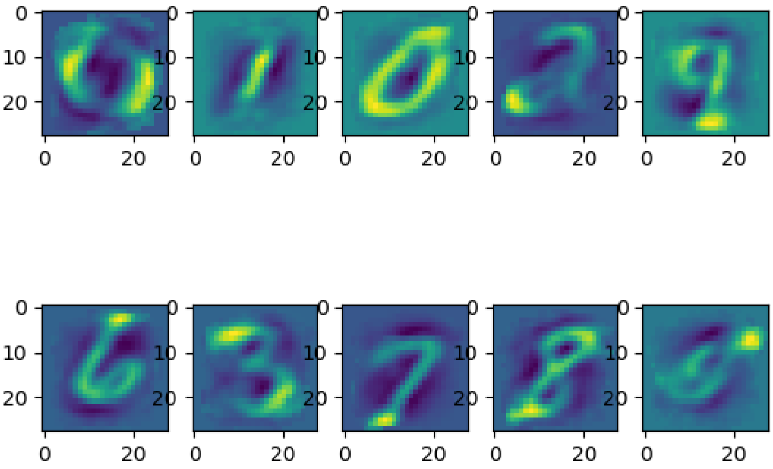

Figure 5.3: Running 700 iterations while randomly initialized .

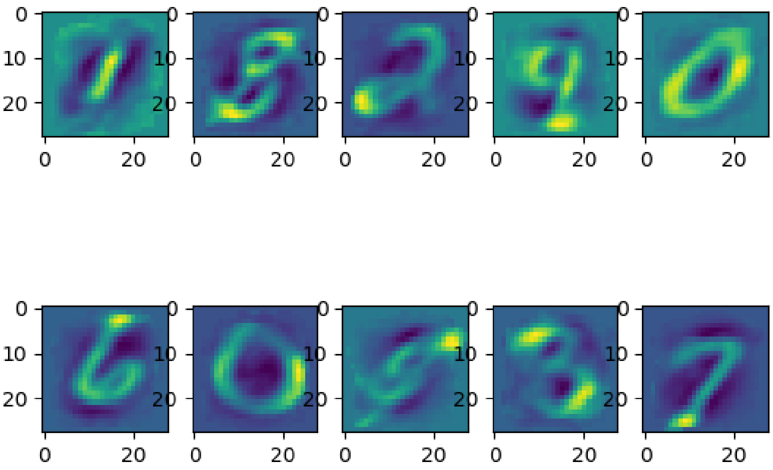

Figure 5.4: Running 3000 iterations while randomly initialized .

It is noticed that, even when initialized to what we think is the golden solution, the results are not exactly what we expected. In fact, in all runs there is no cluster for number 5, while two clusters are representing two version of zero.
(*The code can be found on Github under python directory with name CH5_7.py*)
Solution End

Exercise 5.9 *(Optional) 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:*

$$\begin{aligned}
\pi \sim & \mathit{Dirichlet}_K(\alpha)\\
\eta_k \sim & \mathit{Dirichlet}_V(\beta), \qquad k = 1, \ldots, K\\
z_i \sim & \mathit{Discrete}(\pi), \qquad i = 1, \ldots, N\\
\mathbf{w}_i \sim & \mathit{Multinomial}(\eta_{z_i})
\end{aligned}$$

where N 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:

$$\begin{aligned}
\Pr(w_n = v \mid \mathbb{W}^{(-n)}, z_n = k, \boldsymbol{\beta}) &\propto \#\mathbb{W}_v^{k,(-n)} + \beta_v\\
\Pr(z_n = k \mid \mathbb{Z}^{(-n)}, w_n = v, \mathbb{W}^{(-n)}, \boldsymbol{\alpha}) &\propto (\#\mathbb{Z}_k^{d,(-n)} + \alpha_k) \Pr(w_n = v \mid \mathbb{W}^{(-n)}, z_n = k, \boldsymbol{\beta})
\end{aligned}$$

where:

$$\begin{aligned}
\#\mathbb{W}_v^{k,(-n)} &= \text{number of documents having value } v \text{ among cluster } k \text{ excluding } w_n\\
\#\mathbb{Z}_k^{d,(-n)} &= \text{number of documents in cluster } k \text{ excluding } z_n
\end{aligned}$$

solution End

Exercise 5.10 *(Optional) 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

$$\begin{aligned}
\theta_i \sim &\text{Dirichlet}_K(\alpha), & i = 1, \ldots, N\\
\eta_k \sim &\text{Dirichlet}_V(\beta), & k = 1, \ldots, K\\
z_{i,j} \sim &\text{Discrete}(\theta_i), & j = 1, \ldots, M_i\\
w_{i,j} \sim &\text{Discrete}(\eta_{z_{i,j}}),
\end{aligned}$$

where M_i is the number of words in the jth document. This model is commonly known as Latent Dirichlet Allocation **?**; it is an example of an *admixture* model.

This means that each document is associated with a distribution θ_i 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_i and the η_k. Derive the predictive distributions $p(z_{i,j}|\{z_{\neg i,j}\}, \alpha)$ and $p(w_{i,j}|z_{i,j}, z_{\neg i,j}, w_{\neg i,j}, \beta)$, and hence the conditional distribution $p(z_{i,j}|rest)$*

Solution:
The normalization term for the first probability should be kept because it is dependent on k which is used in the second one

$$\begin{aligned}
\Pr(w_{dn} = v \mid \mathbb{W}^{(-dn)}, z_n = k, \boldsymbol{\beta}) &\quad\propto\quad \frac{\#\mathbb{W}_v^{k,(-dn)} + \beta_v}{\sum_{v_i} \#\mathbb{W}_{v_i}^{k,(-dn)} + \beta_{v_i}}\\
\Pr(z_n = k \mid \mathbb{Z}^{(-dn)}, w_n = v, \mathbb{W}^{(-dn)}, \boldsymbol{\alpha}) &\quad\propto\quad (\#\mathbb{Z}_k^{d,(-dn)} + \alpha_k) \Pr(w_{dn} = v \mid \mathbb{W}^{(-dn)}, z_{dn} = k, \boldsymbol{\beta})
\end{aligned}$$

where:

$$\begin{aligned}
\#\mathbb{W}_v^{k,(-dn)} &\quad=\quad \text{number of words having value } v \text{ among topic } k \text{ excluding } w_{dn}\\
\#\mathbb{Z}_k^{d,(-dn)} &\quad=\quad \text{number of topics having value } k \text{ among document } d \text{ excluding } z_{dn}
\end{aligned}$$

solution End

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?*

The exercise does not require any implementation or derivation, but since I am using python I avoided translating the code to python so I skipped this exercise

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 $\pi \sim \text{Dirichlet}_{100}(10, 10, \ldots, 10)$, *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.*

Solution:

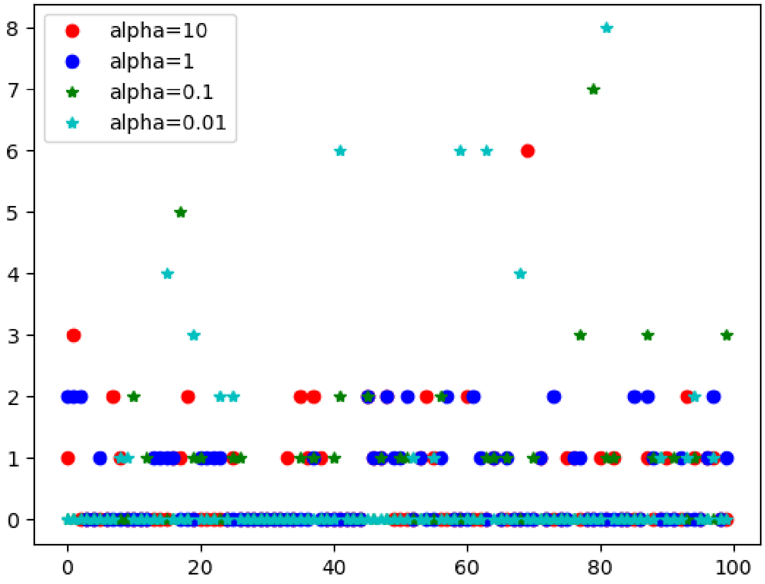

Figure 5.5: Histogram of cluster distribution.

The figure indicates that as α decreases, the clustering is becoming more biased as more samples are belonging to the same cluster.
(*The code can be found on Github under python directory with name CH5_13.py*)

Solution End

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

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

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

Solution:
Part 1:

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

We can start from the moment generating function of a gamma distribution:

$$M_{\gamma_i}(\lambda, \alpha_i, \beta) = \frac{1}{(1 - \frac{\lambda}{\beta})^{\alpha_i}}$$

Using independence, the moment generation function of Z is given by:

$$M_{\sum \gamma_i}(\lambda, \alpha_1, \dots, \alpha_K, \beta) = \prod_{i=1}^{K} M_{\gamma_i}(\lambda, \alpha_i, \beta) = \prod_{i}^{K} \frac{1}{(1 - \frac{\lambda}{\beta})^{\alpha_i}} = \frac{1}{(1 - \frac{\lambda}{\beta})^{\sum_{i=1}^{K} \alpha_i}}$$

Which is a moment generating function of a gamma distribution.

Part 2:

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

Consider the transformation $(\gamma_1, \dots, \gamma_K) = T(\pi_1, \dots, \pi_{K-1}, Z) = (Z\pi_1, Z\pi_2, \dots, Z\pi_{k-1}, Z(1 - \sum_{i=1}^{K-1} \pi_i))$, the Jacobian can be given by:

$$\begin{bmatrix} Z & 0 & \dots & 0 & \pi_1 \\ 0 & Z & \dots & 0 & \pi_2 \\ \dots & \dots & \dots & \dots & \\ & & & & \\ 0 & 0 & \dots & Z & \pi_{K-1} \\ -Z & -Z & \dots & -Z & 1 - \sum_{i=1}^{K-1} \pi_i \end{bmatrix}$$

the determinant of this Jacobian is equal to Z^{K-1}.
The joint distribution of α_i is :

$$p = \prod_{i=1}^{K} \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} e^{-\beta \gamma_i} \gamma_i^{\alpha_i - 1}$$

The joint distribution of the transformed variables is:

$$g = f(T(\alpha))|det| = \underbrace{\prod_{i=1}^{K-1} \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} e^{-\beta(Z\pi_i)} (Z\pi_i)^{\alpha_i - 1}}_{\text{K-1 terms}} \underbrace{\left\{Z(1 - \sum_{i=1}^{K-1} \pi_i)\right\}^{\alpha_K - 1} \frac{\beta^{\alpha_K} e^{-\beta Z(1 - \sum_{i=1}^{K-1} \pi_i)}}{\Gamma(\alpha_K)}}_{\text{k-th term}} \underbrace{Z^{K-1}}_{\text{det}}$$

$$= \frac{(1 - \sum_{i=1}^{K-1} \pi_i)^{\alpha_K - 1} \prod_{i=1}^{K-1} \pi_i^{\alpha_i - 1}}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \beta^{\sum_{i=1}^{K} \alpha_i} e^{-\beta Z} Z^{-1 + \sum_{i=1}^{K} \alpha_i}$$

To get the distribution of pi_is we have to integrate Z:

$$\begin{aligned}
g(\pi) &= \int \frac{(1-\sum_{i=1}^{K-1}\pi_i)^{\alpha_K-1}\prod_{i=1}^{K-1}\pi_i^{\alpha_i-1}}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\beta^{\sum_{i=1}^{K}\alpha_i}e^{-\beta Z}Z^{-1+\sum_{i=1}^{K}\alpha_i}dZ \\
&= \frac{(1-\sum_{i=1}^{K-1}\pi_i)^{\alpha_K-1}\prod_{i=1}^{K-1}\pi_i^{\alpha_i-1}}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\int \beta^{\sum_{i=1}^{K}\alpha_i}e^{-\beta Z}Z^{-1+\sum_{i=1}^{K}\alpha_i}dZ \\
&= \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}(1-\sum_{i=1}^{K-1}\pi_i)^{\alpha_K-1}\prod_{i=1}^{K-1}\pi_i^{\alpha_i-1}
\end{aligned}$$

which results in a Dirichlet distribution for pi_i.

Solution End

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* $(\pi_1+\pi_2,\ldots,\pi_K) \sim$ *Dirichlet*$(\alpha_1+\alpha_2,\alpha_3,\ldots,\alpha_K)$*.*

Solution:
If $(\pi_1,\ldots,\pi_K) \sim \text{Dirichlet}(\alpha_1,\ldots,\alpha_K)$, then we can express the Dirichlet variables in terms of independent Gamma variables as follows (using the previous exercise):

$$(\pi_1,\ldots,\pi_K) = \frac{1}{\sum_{i=1}^{K+1}\gamma_i}(\gamma_1,\ldots,\gamma_K)$$

Working on the Gamma side, the sum of γ_1 and γ_2 is another gamma distribution with $\alpha = \alpha_1+\alpha_2$, then

$$(\pi_1+\pi_2,\ldots,\pi_K) = \frac{1}{\sum_{i=1}^{K+1}\gamma_i}(\gamma_1+\gamma_2,\ldots,\gamma_K) \sim \text{Dirichlet}(\alpha_1+\alpha_2,\alpha_3,\ldots,\alpha_K)$$

Solution End

Exercise 5.16 *Let* $\pi \sim$ *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.*

Solution:
Any such partition is simply summing πs in the interval, and it was proved in the previous exercise that the sum of Dirichlet variables is also Dirichlet.

Solution End
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 $\pi \sim \text{Dirichlet}_K\left(\frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right)$ 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?*

Solution:

Solution:

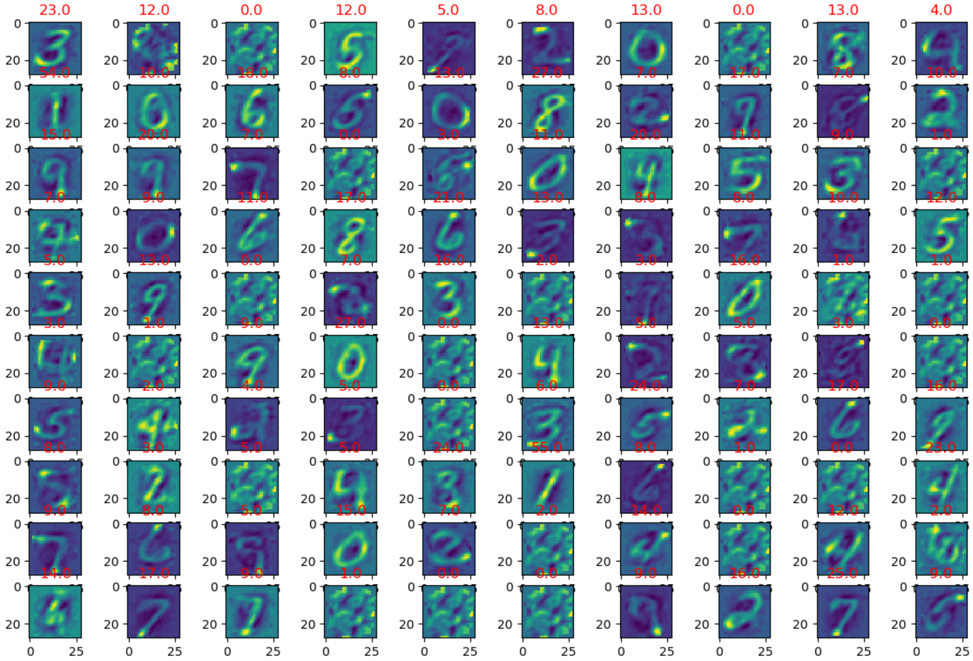

Figure 5.6: Results of clustering with 100 clusters (numbers in red indicate number of samples in a given cluster).

Two main observations can be made here:
1- not all clusters are used, some clusters have zero samples in them
2- multiple clusters contain images for the same digit. For example at least 7 clusters contain 'zero' images and it is clear that zeros are written in different ways in these clusters.
(*The code can be found on Github under python directory with name CH5_17.py*)

Solution End