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?*

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:

$$\begin{aligned} &\mathbf{P}(Z_i = z | X_i, \pi, \mu_1, \sigma^2) \propto P(Z_i = z|\pi) p(X_i|\mu_z, \sigma^2) \\ \text{so,} \qquad &\mathbf{P}(Z_i = 1 | X_i, \pi, \mu_1, \sigma^2) \propto \pi p(X_i|\mu_1, \sigma^2) \\ &\mathbf{P}(Z_i = 0 | X_i, \pi, \mu_1, \sigma^2) \propto P(Z_i = 0|\pi) p(X_0|\mu_z, \sigma^2) \end{aligned}$$

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.

Answer: The regression above was completed, with the X and Y data scaled such that the mean of each set was zero. The optimal β value was calculated as 0.4166, and the distributions are included below.

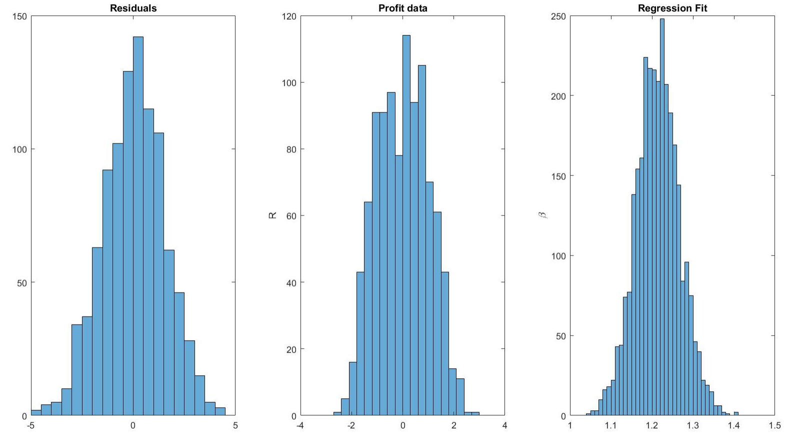


Figure 5.1: Histograms of the residuals, the Y data, and the beta values, with the mean at 0.4166.

The RMSE for the regression is 1.5174.

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.

Answer: For this exercise, sigma was kept at 0.4, inspired by the solution to the previous exercise. It was found that the results were less accurate when sampling sigma. Prior to solving, the data was sorted and whitened.

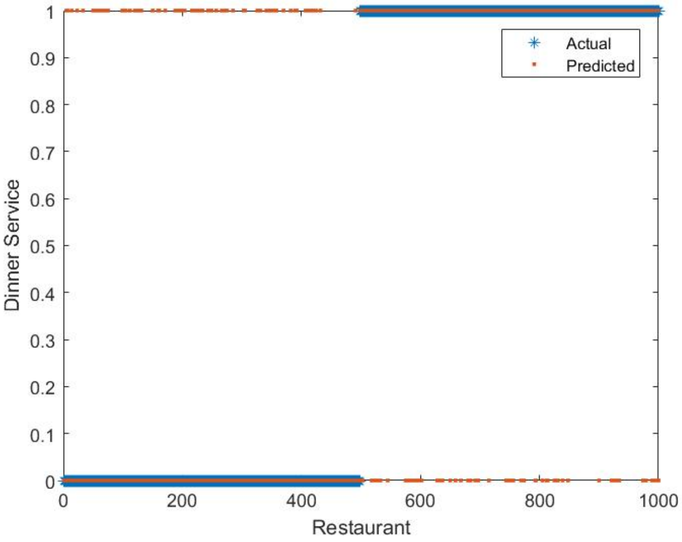


Figure 5.2: Predicted and actual values for the variable Dinner Service.

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*?*

Answer: First, I designate a vector of hyperparameters: $\theta = \{\mu_1, \mu_2, \sigma_1, \sigma_2\}$. The conditional distribution is of the beta-binomial form, which can be derived from the normal likelihood and the posterior.

$$P(Z_i|Z_{-i}, X_i, \theta, \alpha, \beta) = \int P(Z_i|X_i, \pi, \theta) p(\pi|Z_{-i}, \alpha, \beta) d\pi \tag{5.1}$$

$$\propto \int \pi^{Z_i}(1-\pi)^{1-Z_i} N(X_i|\mu_{Z_i}, \sigma^2_{Z_i}) Beta(\pi|\alpha + \sum_{j \neq i} Z_j, \beta + n - \sum_{j \neq i} Z_j) d\pi \tag{5.2}$$

$$\propto N(X_i|\mu_{Z_i}, \sigma^2_{Z_i}) \frac{\Gamma(\alpha_n + \beta_n)}{\Gamma(\alpha_n)\Gamma(\beta_n)} \int \pi^{\alpha_n + Z_i}(1-\pi)^{\beta_n + 1 - Z_i} d\pi \tag{5.3}$$

$$\propto N(X_i|\mu_{Z_i}, \sigma^2_{Z_i}) \frac{\Gamma(\alpha_n + \beta_n)}{\Gamma(\alpha_n)\Gamma(\beta_n)} \frac{\Gamma(\alpha_n + Z_i)\Gamma(\beta_n + 1 - Z_i)}{\Gamma(\alpha_n + \beta_n + 1)} \tag{5.4}$$

$$\propto N(X_i|\mu_{Z_i}, \sigma^2_{Z_i}) Beta - Binom(Z_i|1, \alpha_n, \beta_n) \tag{5.5}$$

$$\propto N(X_i|\mu_{Z_i}, \sigma^2_{Z_i}) \alpha_n^{Z_i} \beta_n^{1-Z_i} \tag{5.6}$$

In the above, $\alpha_n = \alpha + \sum_{j \neq i} Z_j$, and $\beta_n = \beta + n - \sum_{j \neq i} Z_j$. The addition of π in the above enables use of the cluster proportion of the sample.

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?*

Answer: This problem is similar to exercise 2.2, and therefore the solution will be used. Updating the model from the previous exercise, where $\lambda = \sigma^2$, we get:

$$y = \pi_1 N(\mu_1, \lambda) + \pi_2 N(\mu_2, \lambda) \tag{5.7}$$

$$\mu_1, \mu_2 \sim N(0, \lambda) \tag{5.8}$$

$$\lambda \sim Gamma(\alpha, \beta) \tag{5.9}$$

By noting the conjugacy of $\{\lambda_1 \lambda_2, \mu_1, \mu_2\}$, the parameters can be integrated out, by combining this with the solution from exercise 2.2, we get:

$$p(\lambda_1|y) \sim Gamma(\frac{n_1}{2} + \alpha, \beta + \frac{\sum_{i:Z_i=1} y_i^2}{2} - \frac{(\sum_{i:Z_i=1} y_i)^2}{2n_1}) \tag{5.10}$$

$$p(Z_i|Z_{\neg i}, X, \theta) = \frac{\Gamma(\alpha + n_1)\Gamma(\beta + n_2)\Gamma(a_1)\Gamma(a_2)}{\Gamma(\alpha + \beta + n) b_1^{a_1} b_2^{a_2}} \tag{5.11}$$

Where $a_1 = \frac{n_1}{2} + \alpha$, $a_2 = \frac{n_2}{2} + \alpha$, $b_1 = \beta + \frac{\sum_{i:Z_i=1} y_i^2}{2} - \frac{(\sum_{i:Z_i=1} y_i)^2}{2n_1}$, $b_2 = \beta + \frac{\sum_{i:Z_i=0} y_i^2}{2} - \frac{(\sum_{i:Z_i=0} y_i)^2}{2n_1}$.

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.*

Answer: To calculate the cluster proportions, the following terms were added onto the likelihoods in problem 5.2:

$$p(z_i = 1|z_{\neg i}, \alpha) \propto \frac{(N_1 + \alpha)}{N - 1 + \alpha + \beta} N_{pdf}(\mu_1, \sigma^2) \tag{5.12}$$

$$p(z_i = 0|z_{\neg i}, \alpha) \propto \frac{(N_0 + \alpha)}{N - 1 + \alpha + \beta} N_{pdf}(\mu_0, \sigma^2) \tag{5.13}$$

The error between the actual and predicted counts was: $E_0 = 3.21\%$ and $E_1 = 3.19\%$, the plot shown below tracks the counts for each iteration, showing convergence around iteration 10 for both 1 and 0.

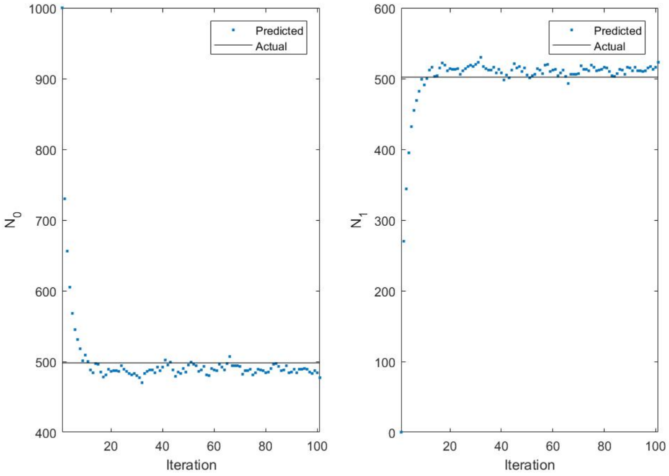


Figure 5.3: Predicted and actual counts for the variable Dinner Service for each iteration.

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

Answer:

$$P(Z_{n+1}|Z_{1:n}, \alpha) = \int P(Z_{n+1}|\pi)p(\pi|Z_{i:n}, \alpha)d\pi = \int n! \prod_{k=1}^{K} \frac{\pi_k^{n_k}}{n_k!} \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k} d\pi \tag{5.14}$$

$$= \frac{\Gamma(\sum_k \alpha_k)n!}{\prod_k \Gamma(\alpha_k)n_k!} \int \prod_{k=1}^{K} \pi_k^{\alpha_k + n_k} d\pi = \frac{\Gamma(\sum_k \alpha_k)n! \prod_k \Gamma(\alpha_k + n_k)}{\prod_k \Gamma(\alpha_k)n_k! \Gamma(\sum_k \alpha_k + n)} \tag{5.15}$$

The above represents the multivariate Dirchlet, $DM(n, \alpha_n)$, where n_k is the number of points in $Z_{1:n}$ which belong to k, and $\alpha_n = (\alpha_1 + n_i, ..., \alpha_k + n_k)$.

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.*

Answer: The generated clusters are given in the plot below:

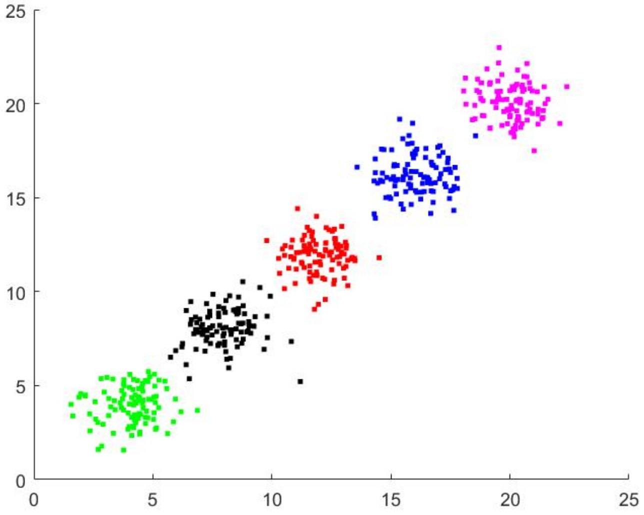


Figure 5.4: Generated clusters.

The Gibbs sampler from above was modified for $K = 5$ clusters, where each cluster had 100 data points and the sampler had 10 iterations. The total wrong number of assignments was 2, and the plot below shows the sorted z variable, where the 2 inaccuracies are readily observed.

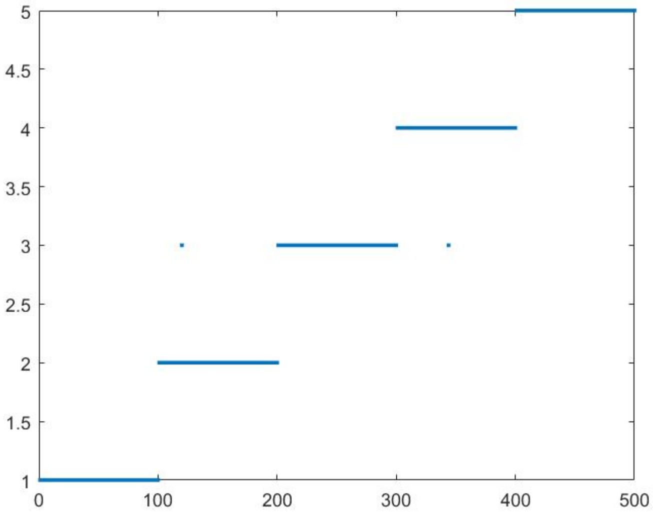


Figure 5.5: Sorted assignments of latent variable.

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.*

Answer: PCA was done to reduce each image to 50 entities. The figure below shows the change in image quality before and after PCA.

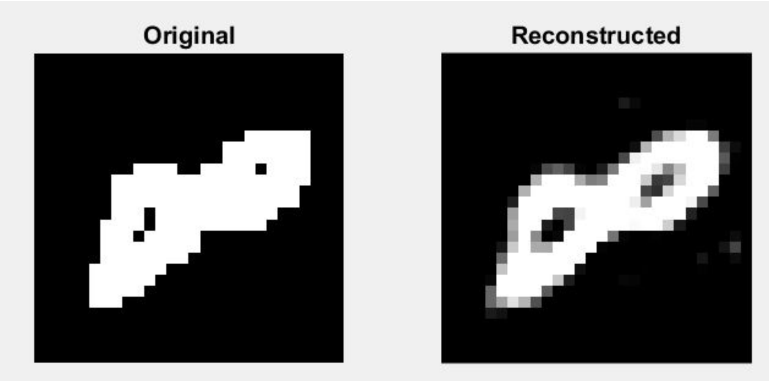


Figure 5.6: Digit before and after PCA with 50 components.

As shown, most of the image is retained using PCA, however, the edges are more blurry with PCA and therefore may lead to vectors with means that are less-distinct from one another.

The Gibbs sampler implemented in exercise 5.7 was not very successful in classifying the digits, the plot below shows the true labels and the predicted labels.

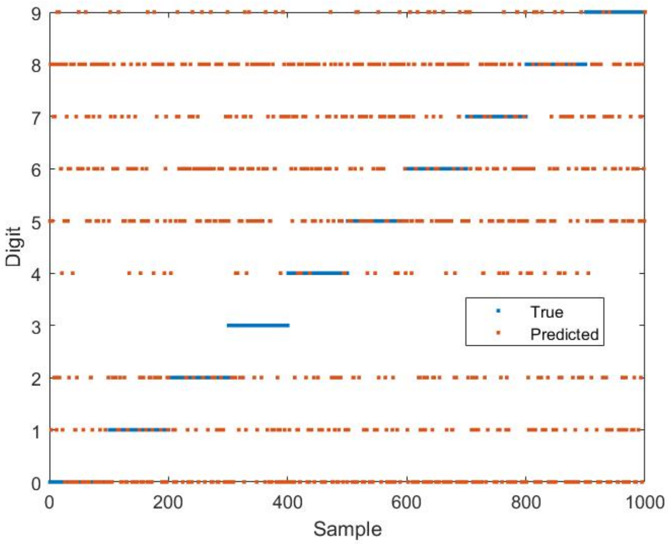


Figure 5.7: Labels solved for by the Gibbs Sampler.

It is clear that the sampler is doing something, because there are groupings of labels throughout, but it is likely that the labels are being mis-assigned. This is reasonable since many of the digits looks similar, and also because the value of Σ was kept at the identity matrix. The histogram of labels below shows the disproportionate assignments.

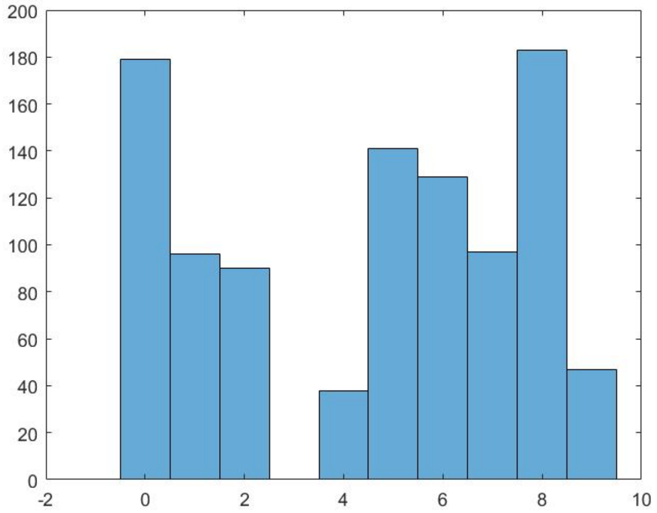


Figure 5.8: Histogram of the labels determined by the Gibbs Sampler.

Lastly, the percent error in label assignment was calculated to be 89%, showing that the Gibbs sampler with fixed Σ and only 10 clusters is not the best way to classify the digits.

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:*

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

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.

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

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

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 $p(z_{i,j}|rest)$*

Answer:

$$Joint = \prod_{i=1}^{N} p(\omega_i|z_i, \eta)p(z_i|\theta_i)p(\theta_i) \prod_{k=1}^{K} p(n_k) \tag{5.16}$$

$$= \prod_{i=1}^{N} p(\theta_i) \prod_{j=1}^{M} p(\omega_{ij}|z_{ij}, \eta)p(z_{ij}|\theta_i) \prod_{k=1}^{K} p(\eta_k) \tag{5.17}$$

$$p(z, w) = \prod_{i=1}^{N} \int p(\theta_i) \prod_{j=1}^{M_i} p(z_{ij}|\theta_i)d\theta_i \prod_{k=1}^{K} \int p(\eta_k)p(\omega_{ij} z_{ij}, \eta)d\eta_k \tag{5.18}$$

$$= \prod_{i=1}^{N} Dur^K_{mult}(n_i^K|\alpha) \prod_{k=1}^{K} Dir^V_{mult}(n_k^V|\beta) \tag{5.19}$$

$$p(z_{ij} = k|\omega_{ij}) \propto p(z_{ij} = k|z_{i\neg j})p(\omega_{ij} = v|z, \omega_{ij}) \tag{5.20}$$

$$\propto Dir_{mult}(z_{ij}|n^K_{i\neg j} + \alpha)Dir_{mult}(\omega_{ij}|n^V_{k\neg ij} + \beta) \tag{5.21}$$

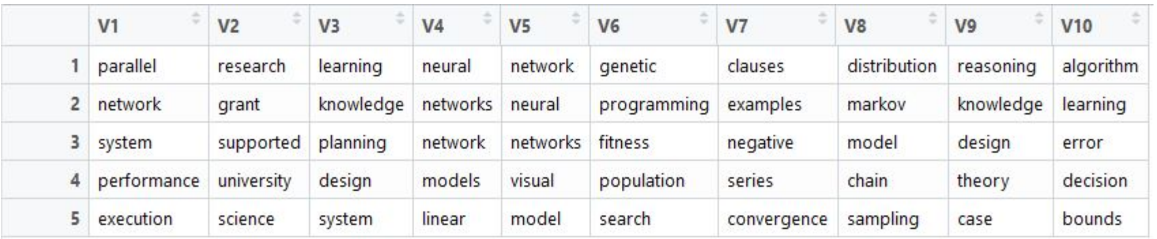
$$\propto \frac{n_{ik} + \alpha_k}{\sum_k n_{ik} + \alpha_k} \frac{n_{kv} + \beta_v}{\sum_v n_{kv} + \beta_v} \tag{5.22}$$

$$\propto (n_{ik} + \alpha_k) \frac{n_{kv+\beta_v}}{\sum_v n_{kv} + \beta_v} \tag{5.23}$$

In the above, n^K_{ik} refers to the number of words in cluster k in document i (cluster assignment count), and n^V_{kv} is the number of times in the collection that word v is in cluster k (word distribution of topic).

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 above function was used with the `lda` package, and the results are below:



	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
1	parallel	research	learning	neural	network	genetic	clauses	distribution	reasoning	algorithm
2	network	grant	knowledge	networks	neural	programming	examples	markov	knowledge	learning
3	system	supported	planning	network	networks	fitness	negative	model	design	error
4	performance	university	design	models	visual	population	series	chain	theory	decision
5	execution	science	system	linear	model	search	convergence	sampling	case	bounds

Figure 5.9: Top words in each cluster.

This model is called a topic model because it helps to discover the abstract topics occuring in a collection of documents, the above model uses text mining to define 10 topics, which each have most probable words.

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 $\pi \sim \text{Dirichlet}_{100}(10, 10, \dots, 10)$*, and then sample 10 cluster indicators* $z_i \sim \pi$*. Record the list of cluster indicators, e.g.* $\{1, 10, 11, 11, \dots\}$*. 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.*

Answer: For the alpha values $\{0.01, 0.1, 1.0, 10\}$, the number of clusters was $\{15, 30, 36, 40\}$. It appears that increasing the alpha leads to more clusters.

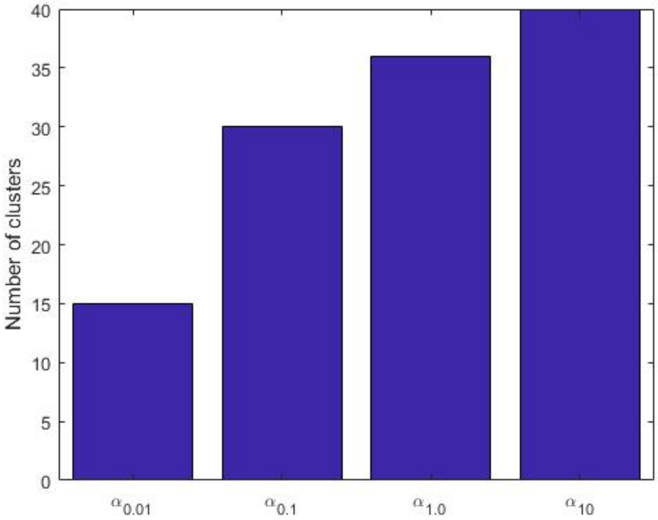


Figure 5.10: Cluster size vs α.

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

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

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.*

Answer: Using convolution:

$$f_{x+y}(z) = \int_0^z f_x(x) f_y(z-x) dx \tag{5.24}$$

$$= \int_0^z \frac{\beta^{\alpha_x}}{\Gamma(\alpha_x)} x^{\alpha_x - 1} e^{-\beta x} \frac{\beta^{\alpha_y}}{\Gamma(\alpha_y)} (z-x)^{\alpha_y - 1} e^{-\beta(z-x)} dx \tag{5.25}$$

$$= \frac{\beta^{\alpha_x + \alpha_y}}{\Gamma(\alpha_x)\Gamma(\alpha_y)} e^{-\beta z} \int_0^z x^{\alpha_x - 1} (z-x)^{\alpha_y - 1} dx \tag{5.26}$$

Using a change of variables with $x = zp$:

$$= \frac{\beta^{\alpha_x + \alpha_y}}{\Gamma(\alpha_x)\Gamma(\alpha_y)} e^{-\beta z} \int_0^z (zp)^{\alpha_x - 1} (z(1-p))^{\alpha_y - 1} z dp \tag{5.27}$$

$$= \frac{\beta^{\alpha_x + \alpha_y}}{\Gamma(\alpha_x)\Gamma(\alpha_y)} e^{-\beta z} z^{\alpha_x - 1} z^{\alpha_y - 1} z \int_0^z p^{\alpha_x - 1} (1-p)^{\alpha_y - 1} dp \tag{5.28}$$

$$= \frac{\beta^{\alpha_x + \alpha_y}}{\Gamma(\alpha_x)\Gamma(\alpha_y)} e^{-\beta z} z^{\alpha_x + \alpha_y - 1} \left[\frac{\Gamma(\alpha_x)\Gamma(\alpha_y)}{\Gamma(\alpha_x + \alpha_y)} \right] \tag{5.29}$$

$$= \frac{\beta^{\alpha_x + \alpha_y}}{\Gamma(\alpha_x + \alpha_y)} e^{-\beta z} z^{\alpha_x + \alpha_y - 1} \tag{5.30}$$

$$= Gamma(\alpha_x + \alpha_y, \beta) \tag{5.31}$$

We also note that the sum of N Gamma variables is also a Gamma distribution.

$$\pi_i = \frac{\gamma_i}{Z} \rightarrow \gamma_i = \pi_i Z = \pi_i \pi_K \; \forall i \neq K \tag{5.32}$$

$$Z = \pi_K = \sum_i \gamma_i \tag{5.33}$$

If $\pi_K = \sum_i \gamma_i$, then $\gamma_k = \pi_K - \sum_i^{K-1} \gamma_i = \pi_K(1 - \sum_i^{K-1} \pi_i)$

Taking the determinant of the Jacobian: $|J| = \left|\frac{\partial \gamma}{\partial \pi}\right|$ gives the result π_K^{K-1}. Using this result to transform the probability density gives:

$$p_\pi(\pi) = |J| p_\gamma(\gamma) \text{ (change of variables)} \tag{5.34}$$

$$(\pi_K^{K-1}) \left[\frac{\beta^{\sum_i \alpha_i}}{e^{\beta \sum_i \gamma_i}} \right] \prod_{i=1}^{K} \frac{\gamma_i^{\alpha_i - 1}}{\Gamma(\alpha_i)} \tag{5.35}$$

Separating i from 1 to $K-1$, and $i = K$ and simplifying yields:

$$\pi_K^{K-1} \frac{\beta^{\sum_i \alpha_i}}{e^{\beta \pi_K}} \prod_{i=1}^{K-1} \left[\frac{(\pi_i \pi_K)^{\alpha_i - 1}}{\Gamma(\alpha_i)} \right] \frac{\pi_K^{\alpha_K - 1}}{\Gamma(\alpha_K)} \left(1 - \sum_{i=1}^{K-1} \pi_i \right)^{\alpha_K - 1} \tag{5.36}$$

Integrating out π_K yields:

$$\int p_\pi(\pi)d\pi_K = \left(1 - \sum_{i=1}^{K-1}\right)^{\alpha_k - 1} \frac{\Gamma(\sum_i \alpha_i)}{\Gamma(\alpha_K)} \prod_{i=1}^{K-1} \frac{\pi^{\alpha_i - 1}}{\Gamma(\alpha_i)} = Dir(\pi_1 \dots \pi_{K-1}, 1 - \sum_{i=1}^{K-1} \pi_i | \alpha_1 \dots \alpha_K) \tag{5.37}$$

You will probably find this relationship helpful in proving the following

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

Answer: We know that $\pi \sim Dir(\alpha)$ and that $\gamma_i \sim Gamma(\alpha_i, \beta)$, and that $\pi_1 = \frac{\gamma_1}{Z}, \pi_2 = \frac{\gamma_2}{Z}$.

Summing together, we get that $\pi_1 + \pi_2 = \frac{\gamma_1 + \gamma_2}{Z} \to Gamma(\alpha_1 + \alpha_2, \beta) \to Dir(\alpha_1 + \alpha_2, \alpha_3 \dots \alpha_K)$.

Since $Z = \gamma_1 + \gamma_2 + \sum_{i=3}^{K} \gamma_i$.

Exercise 5.16 *Let $\pi \sim Dirichlet_K\left(\frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right)$, 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.*

Answer: In the previous exercise, we saw that summing α enables the grouping of the corresponding components. Weights which are multiples of $1/k$ contain multiple π_k values, which means that their sum including the rest of the weights is also a distributed Dirichlet.

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 $A_1, \dots, A_K$ of Ω, the masses assigned to $A_1, \dots, A_k$ are distributed according to a Dirichlet $(\alpha H(A_1), \dots, \alpha H(A_K))$ 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.

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?*

Answer: The observations were assigned to 100 clusters prior to solving the problem, and the plots below show the distribution of clusters per iteration. The clusters are not close to what we expect, given that there

[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.

are certain labels which are represented more than others, i.e. the distribution is not flat (and in some cases, not close to flat)

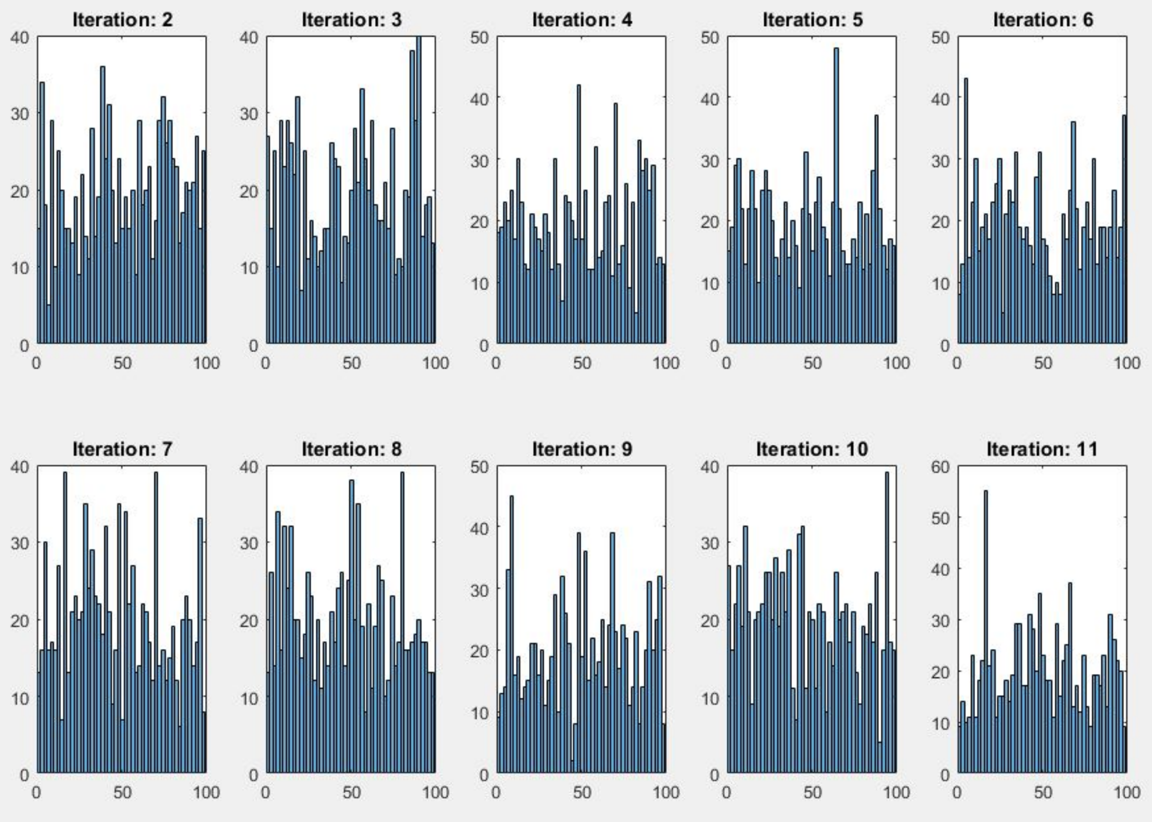


Figure 5.11: Cluster distribution for each iteration.