Section 5: Mixture models

5.1 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?*

Proof: The result is shown in Figure 5.6. Before Bayesian regression, the raw data is not Gaussian. But the Bayesian regression, the residual is normal. ■

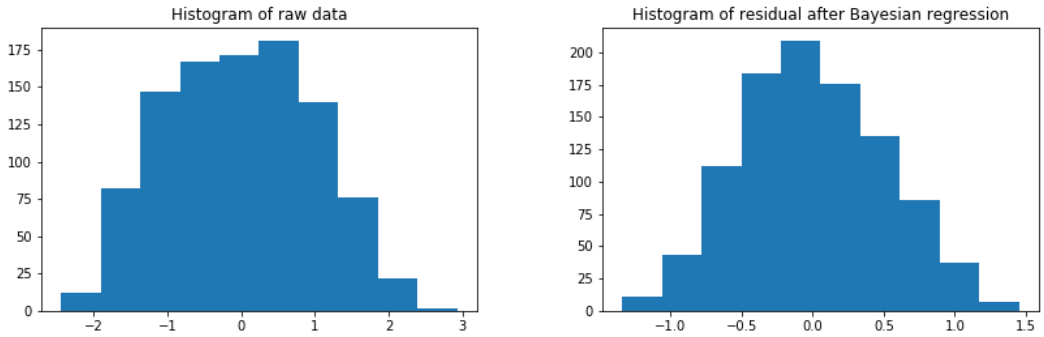


Figure 5.1: Comparison of residuals before and after Bayesian regression

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:

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

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.

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.

Proof:

We use the last sample to get the label for each data. We can see from Table 5.1 that around 85 percentage of data are correctly clustered. We can also plot the histograms of μ_1 and μ_2 which are shown in Figure 5.3.

Ture/Predicted	0	1
0	429	69
1	79	423

Table 5.1: True vs Predicted cluster

■

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 $P(Z_i|Z_{\neg i}, X_i, \mu_1, \mu_2, \sigma_1, \sigma_2, \alpha, \beta)$, where $Z_{\neg i}$ means all the values of Z except Z_i?*

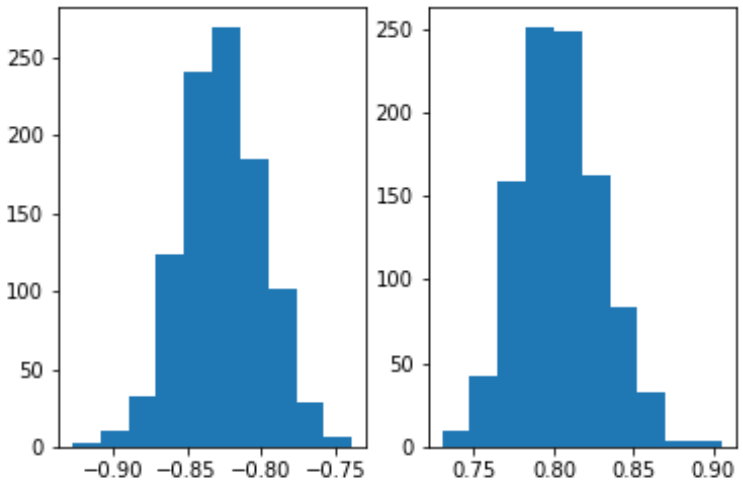


Figure 5.2: Histogram

Proof: We treat σ_1 and σ_0 as hyperparameters.

$$\begin{aligned} P(Z_i|Z_{\neg i}, X_i, \mu_1, \mu_0, \sigma_1, \sigma_0, \alpha, \beta) &\propto p(X_i|Z_i, else)p(Z_i|Z_{\neg i}) \\ &\propto p(X_i|Z_i, else)\int p(Z_i|\pi)p(\pi|Z_{\neg i})d\pi \\ &\propto N(X_i|\mu_{Z_i}, \sigma^2_{Z_i})Beta(\alpha + n_1 + Z_i, \beta + n_0 + 1 - Z_i) \end{aligned} \tag{5.1}$$

■

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?*

Proof: For simplicity I treat σ_1^2 and σ_2^2 as hyperparameters. And give μ_1 and μ_2 a $N(0, 100)$ prior. Then $X_i|Z_i \sim N(0, \sigma^2_{Z_i} + 100)$. Therefore,

$$\begin{aligned} P(Z_i|Z_{\neg i}, X_i, \theta) &\propto p(X_i|Z_i)p(Z_i|Z_{\neg i}) \\ &\propto p(X_i|Z_i)\int p(Z_i|\pi)p(\pi|Z_{\neg i})d\pi \\ &\propto N(X_i|0, \sigma^2_{Z_i} + 100)Beta(\alpha + n_1 + Z_i, \beta + n_0 + 1 - Z_i). \end{aligned} \tag{5.2}$$

■

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.*

Proof: I implemented the fully uncollapsed Gibbs sampler. Suppose we have data $x_1, ..., x_n$ coming from the model $g(x|\theta) = wN(x|\mu_1, \sigma_1^2) + (1 - w)N(x|\mu_2, \sigma_2^2)$. If we introduce a latent variable d_i for each x_i

and define $w_1 = w$, and $w_2 = 1 - w$, then the likelihood can be augmented into $g(x_1, ..., x_n, d_1, ..., d_n|\theta) = \prod_{i=1}^{n} w_{d_i} N(x_i|\mu_{d_i}, \sigma^2_{d_i})$. Set the priors as $f(\mu_1) = N(0, 100)$, $f(\mu_2) = N(0, 100)$, $f(\sigma^2_d) = InvGamma(1, 1)$, and $f(w)$ is uniform on $(0, 1)$, then we have the joint distribution:

$$g(x_1, ..., x_n, d_1, ..., d_n, \theta) = f(\sigma^2)f(\mu_1)f(\mu_2)f(w)\prod_{i=1}^{n} w_{d_i} N(x_i|\mu_{d_i}, \sigma^2_{d_i}).$$

The Gibbs sampler can be derived as follows(suppose $n_1 = \sum_i 1_{\{d_i=1\}}$ and $n_2 = \sum_i 1_{\{d_i=2\}}$), for the $(k+1)$th iteration:

$$w^{(k+1)}|\text{rest} \sim Beta(1 + n_1^{(k)}, 1 + n_2^{(k)}),$$

$$\mu_1^{(k+1)}|\text{rest} \sim N(\frac{\sum_{i:d_i^{(k)}=1} x_i}{(\sigma_1^2)^{(k)}/100 + n_1^{(k)}}, \frac{1}{1/100 + n_1^{(k)}/(\sigma_1^2)^{(k)}}),$$

$$\mu_2^{(k+1)}|\text{rest} \sim N(\frac{\sum_{i:d_i^{(k)}=2} x_i}{(\sigma_2^2)^{(k)}/100 + n_2^{(k)}}, \frac{1}{1/100 + n_2^{(k)}/(\sigma_2^2)^{(k)}}),$$

$$P(d_i^{(k+1)} = d|\text{rest}) = \frac{w_d^{(k+1)} N(x_i|\mu_d^{(k+1)}, (\sigma_d^2)^{(k)})}{w^{(k+1)} N(x_i|\mu_1^{(k+1)}, (\sigma_1^2)^{(k)}) + (1 - w^{(k+1)}) N(x_i|\mu_2^{(k+1)}, (\sigma_2^2)^{(k)})},$$

$$(\sigma_1^2)^{(k+1)}|\text{rest} \sim InvGamma(1 + n_1/2, 1 + (\sum_{i:d_i^{(k+1)}=1} (\mu_1^{(k+1)} - x_i)^2)/2),$$

$$(\sigma_2^2)^{(k+1)}|\text{rest} \sim InvGamma(1 + n_2/2, 1 + (\sum_{i:d_i^{(k+1)}=2} (\mu_2^{(k+1)} - x_i)^2)/2).$$

We use the last sample to get the label for each data. We can see from Table 5.2 that around 80 percentage of data are correctly clustered. We can also plot the histograms of samples of parameters and the histogram of samples generated with sample parameters which are shown in Figure 5.3.

Ture/Predicted	0	1
0	340	36
1	158	466

Table 5.2: True vs Predicted cluster

■

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:

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

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. Here, α is a D-dimensional vector where $\alpha_k > 0$ and $\sum_k \alpha_k \geq 1$. The expectation of a Dirichlet distribution is given by the normalized parameter vector, $E[\pi] = \frac{(\alpha_1, ..., \alpha_K)}{\sum_k \alpha_k}$. The absolute magnitude of the parameter acts like an inverse variance: the smaller its values, the further a given sample is from the expected value.

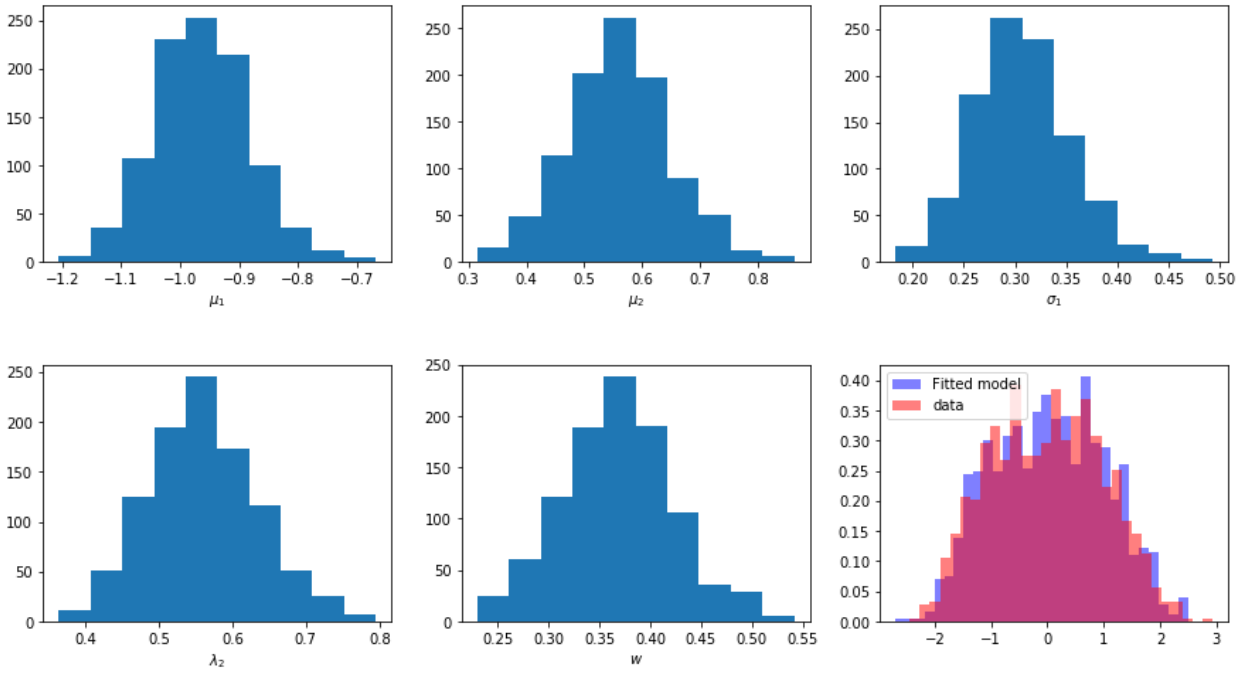


Figure 5.3: Histograms

Exercise 5.6 *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* $E[\pi] = \frac{(\alpha_1,\ldots,\alpha_K)}{\sum_k \alpha_k}$.

Proof:

$$p(\pi|Z_{1:n}) \propto \prod_{k=1}^{K} \pi_k^{\alpha_k+n_k-1},$$

where $n_k = \sum_i 1_{Z_i=k}$. Therefore, The posterior is Dirichlet$(\alpha_1 + n_1, ..., \alpha_K + n_K)$.

$$\begin{aligned} p(Z_{n+1} = j|Z_{1:n}) &= \int f(Z_{n+1} = j|\pi)f(\pi|Z_{1:n})d\pi \\ &\propto \int \pi_j \prod_{k=1}^{K} \pi_k^{\alpha_k+n_k-1} d\pi \\ &\propto \alpha_j + n_j \end{aligned} \tag{5.3}$$

■

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.*

Proof: The code is in file 7and8.py. I generated a mixture of Gaussian with five components. And run the Gibbs sampler and use the final sample of cluster indicator as my cluster prediction. The result is shown in Figure 5.4. We can see that we get quite reasonable results.

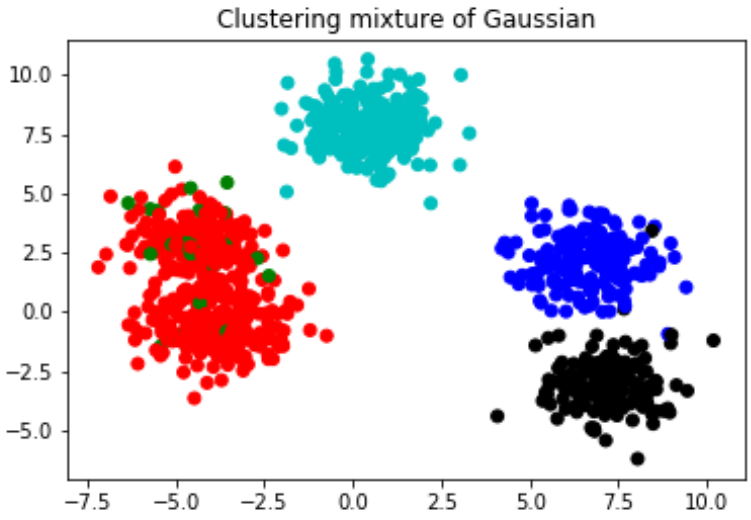


Figure 5.4

■

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.

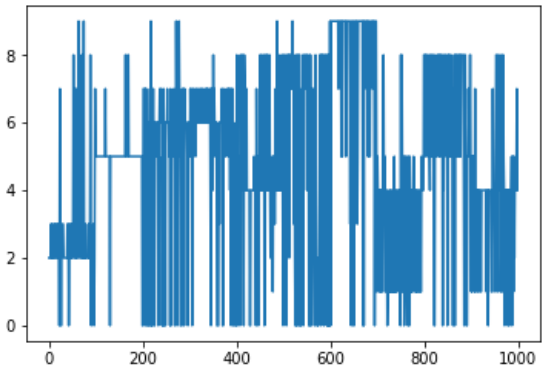
Now, try running your Gibbs sampler with 10 classes, and $\alpha_1 = \alpha_2 = \cdots = \alpha_{10} = 1$. *This prior corresponds to a uniform distribution on the 9-simplex. It's fine to use a spherical covariance here... in fact it will work fine if you just have a prior on the means, and fix* $\sigma^2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

Here are some ways you can visualize your output:

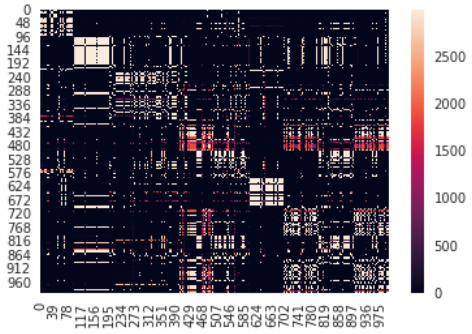
- *Based on a single sample, plot the recovered clustering vs the ground truth clustering.*
- *Based on a single sample, visualize the mean image for each cluster, by multiplying the mean embedding with the coefficients obtained using PCA.*

- *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.*

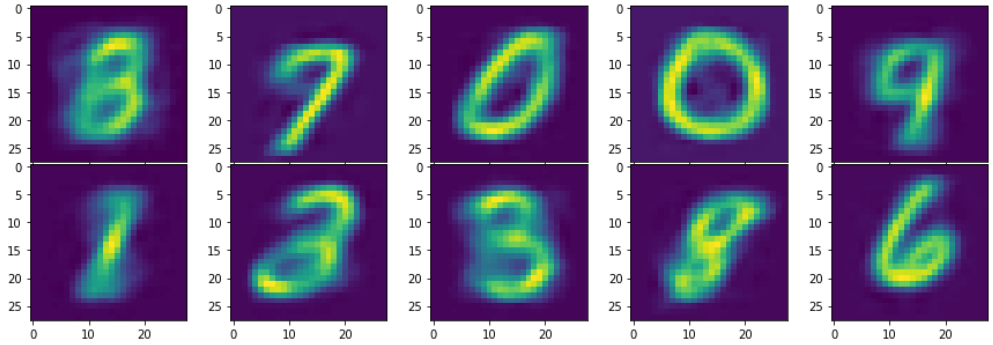
Proof: We ran the algorithm for dimension reducted MNIST data. After 2000 iterations, we get one cluster sample which is plotted in Figure 5.5a. In Figure 5.5c, we plotted the co-occurrence matrix for the images. We can see that there are certain block structure shown in both pictures. We also plotted the average images within each clusters in Figure **??**. ■



(a) Recovered clustering vs true cluster



(b) Heatmap of co-occurrence matrix



(c) Average images for each cluster

Figure 5.5

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:*

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

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.

Proof: Collapsed Gibbs sampler:

$$p(z_i|z_{\neg i}, \mathbf{w}) \propto p(z|\mathbf{w}) \propto p(\mathbf{w}|z)p(z) \propto p(\mathbf{w}|z)p(z_i|z_{\neg i}).$$

From Ex 5.6, we know that

$$p(z_i|z_{\neg i}) \propto \alpha_{z_i} + n_{z_i}$$

where n_k is the number of documents (exclude i) that belongs to kth group.

Now, consider:

$$\begin{aligned} p(\mathbf{w}|z) &= \int p(\mathbf{w}, \eta, z)/p(z)d\eta = \int p(\mathbf{w}|\eta, z)p(\eta)d\eta \\ &= \prod_{k=1}^{K} \int p(\eta_k)p(\mathbf{w}_i|\eta_k)^{1_{z_i=k}} \prod_{j:z_j=k, j\neq i} p(\mathbf{w}_j|\eta_k, z_j = k)d\eta_k \end{aligned} \tag{5.4}$$

Let's denote:

$$A_k = \int p(\eta_k) \prod_{j:z_j=k, j\neq i} p(\mathbf{w}_j|\eta_k, z_j = k)d\eta_k$$

$$B_k = \int p(\eta_k)p(\mathbf{w}_i|\eta_k, z_i = k) \prod_{j:z_j=k, j\neq i} p(\mathbf{w}_j|\eta_k, z_j = k)d\eta_k$$

Then $p(\mathbf{w}|z) = B_{z_i}/A_{z_i} \prod_{k=1}^{K} A_k \propto B_{z_i}/A_{z_i}$(consider z_i as the only variable.)

Then $p(z_i|z_{\neg z}, \mathbf{w}) \propto (\alpha_{z_i} + n_{z_i})B_{z_i}/A_{z_i}$.

$$\begin{aligned} B_k/A_k &= \frac{\int p(\eta_k)p(\mathbf{w}_i|\eta_k, z_i = k) \prod_{j:z_j=k, j\neq i} p(\mathbf{w}_j|\eta_k, z_j = k)d\eta_k}{\int p(\eta_k) \prod_{j:z_j=k, j\neq i} p(\mathbf{w}_j|\eta_k, z_j = k)d\eta_k} \\ &\propto \frac{\int \prod_{v=1}^{V} \eta_{k,v}^{\beta_v - 1} \prod_{j:z_j=k, or j=i} (\prod_{v=1}^{V} \eta_{k,v}^{w_{j,v}})d\eta_k}{\int \prod_{v=1}^{V} \eta_{k,v}^{\beta_v - 1} \prod_{j:z_j=k, j\neq i} (\prod_{v=1}^{V} \eta_{k,v}^{w_{j,v}})d\eta_k} \\ &\propto \frac{\prod_{v=1}^{V} \Gamma(\beta_v + \sum_{j:z_j=k, or j=i} w_{j,v}) \Gamma(\sum_{v=1}^{V} (\beta_v + \sum_{j:z_j=k, j\neq i} w_{j,v}))}{\prod_{v=1}^{V} \Gamma(\beta_v + \sum_{j:z_j=k, j\neq i} w_{j,v}) \Gamma(\sum_{v=1}^{V} (\beta_v + \sum_{j:z_j=k, or j=i} w_{j,v}))} \end{aligned} \tag{5.5}$$

■

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.*

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

Proof: Both $p(z_{i,j}|\{z_{\neg i,j}\}, \alpha)$ and $p(w_{i,j}|z_{i,j}, z_{\neg i,j}, w_{\neg i,j}, \beta)$ are Categorical-Dirichlet. The collapsed gibbs sampler is the following:

$$\begin{aligned}
p(z_{i,j} = k|z_{i,-j}, w_{i,j} = v) \propto & p(z_{i,j} = k|z_{i,-j})p(w_{i,j} = v|\{w_{i,j} : z_{i,j} = k\}) \\
= & \frac{m_{i,k}^{-j} + \alpha_k}{M_i - 1 + \sum_k \alpha_k} \cdot \frac{\rho_{k,v}^{-w_{i,j}} + \beta_v}{\sum_{v'}(\rho_{k,v'}^{-w_{i,j}} + \beta_{v'})} \\
\propto & (m_{i,k}^{-j} + \alpha_k) \cdot \frac{\rho_{k,v}^{-w_{i,j}} + \beta_v}{\sum_{v'}(\rho_{k,v'}^{-w_{i,j}} + \beta_{v'})}
\end{aligned}$$

■

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?*

Proof: In Table 5.3, the top 5 words are listed for 7 clusters. We can see that the words within each cluster are meaningfully connected, i.e., forming a meaningful topic. This is why this kind of model is called topic model.

■

	0	1	2	3	4	5	6
genetic	neural	research	bayesian	algorithm	decision	theory	design
search	network	grant	markov	learning	learning	logic	knowledge
programming	networks	university	data	bounds	training	belief	reasoning
evolutionary	training	report	models	class	algorithm	revision	system
fitness	recurrent	science	distribution	bound	data	causal	case

Table 5.3: Top words for cora dataset

	0	1	2	3	4	5	6	7	8	9
0	42	55	33	14	95	1	75	2	90	43
1	97	88	10	97	49	84	38	4	87	92
2	27	73	5	55	96	59	89	18	62	60
3	44	31	12	89	81	41	70	45	81	81
4	17	72	16	35	45	5	52	13	45	99

Table 5.4: $\alpha = 1$

5.2 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.*

Proof: In this following tables and figures, samples of cluster indicators have been listed for different α. We can see as α increases, the clusters collapse, i.e., only a small number of clusters are nonempty.

■

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

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

	0	1	2	3	4	5	6	7	8	9
0	2	18	43	2	62	30	2	71	18	2
1	12	34	49	24	96	9	68	33	12	12
2	64	36	28	58	28	23	36	30	52	36
3	3	33	80	3	7	81	30	33	95	30
4	18	64	64	68	72	2	18	56	53	94

Table 5.5: $\alpha = 0.1$

	0	1	2	3	4	5	6	7	8	9
0	89	49	44	59	44	59	44	14	89	59
1	75	77	77	77	75	77	77	77	40	77
2	54	15	54	54	32	32	32	45	54	15
3	44	44	98	44	44	63	44	63	44	44
4	90	90	21	90	90	21	28	21	90	21

Table 5.6: $\alpha = 0.01$

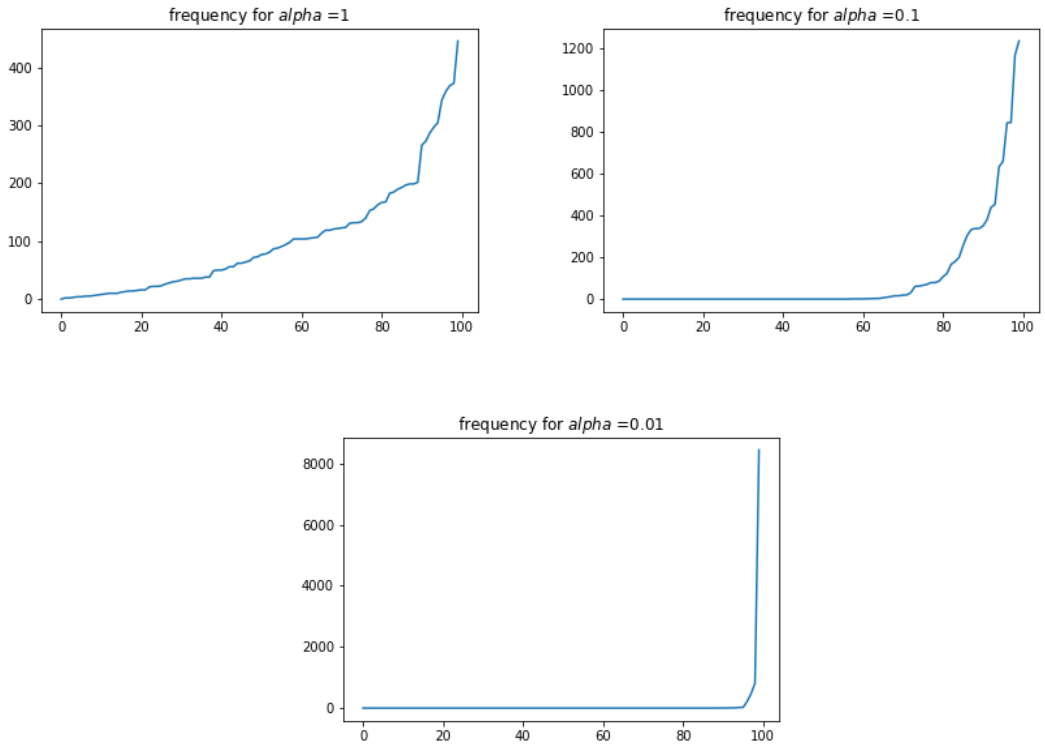


Figure 5.6: Comparison of residuals before and after Bayesian regression

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.*

Proof: To show $Z \sim Gamma(\sum \alpha_i, ..., \beta)$, we can use moment generating function. $E(e^{tX}) = \frac{\beta^\alpha \Gamma(\alpha)}{\Gamma(\alpha)(\beta - t)^\alpha}$, if $X \sim Gamma(\alpha, \beta)$. Then the argument is obvious.

To show that $\pi = \left(\frac{\gamma_1}{Z}, \ldots, \frac{\gamma_K}{Z}\right) \sim \text{Dirichlet}(\alpha_1, \alpha_K)$, we define $X = (\gamma_1, ..., \gamma_K)$ and $Y = (\pi_1, \pi_{K-1}, Z)$. The the map between X and Y is one-to-one.

Consider

$$\frac{dX}{dY} = \begin{bmatrix} z & 0 & \cdots & 0 & \pi_1 \\ 0 & z & \cdots & 0 & \pi_2 \\ & & \cdots & & \\ 0 & 0 & \cdots & z & \pi_{K-1} \\ -z & -z & \cdots & -z & 1 - \pi_1 - \cdots - \pi_{K-1} \end{bmatrix}.$$

The determinant of $\frac{dX}{dY}$ is easy since adding one row to another doesn't change the determinant, so we can add every row to the last row. Then the determinant is obvious: $\frac{dX}{dY} = z^{K-1}$.

Therefore(denote $\pi_K = 1 - \pi_1 - \cdots - \pi_{K-1}$),

$$\begin{aligned} f(Y) &= f(X)|det(\frac{dX}{dY})| \\ &= \prod_{k=1}^{K} \frac{\beta^{\alpha_k}}{\Gamma(\alpha_k)} (\pi_k z)^{\alpha_k - 1} e^{-\beta \pi_k z} z^{K-1} \\ &= \prod_{k=1}^{K} \frac{\beta^{\alpha_k}}{\Gamma(\alpha_k)} \pi_k^{\alpha_k - 1} e^{-\beta z} z^{\sum \alpha_k - 1} \end{aligned} \tag{5.6}$$

We can see that $\pi_1, ..., \pi_{K-1}$ is independent of z, and the marginal of $\pi_1, ...\pi_{K-1}$ (equivalently $\pi_1, ...\pi_{K-1}, \pi_K$) is Dirichlet with $(\alpha_1, ...\alpha_K)$.

■

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

Proof: If we have $\gamma_i \overset{\text{iid}}{\sim} \text{Gamma}(\alpha_i, \beta)$, then $\pi = \left(\frac{\gamma_1}{Z}, \ldots, \frac{\gamma_K}{Z}\right) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_K)$. Then $\gamma_1 + \gamma_2 \sim Gamma(\alpha_1 + \alpha_2, \beta)$. So $((\gamma_1 + \gamma_2)/Z, \gamma_3/Z, ..., \gamma_K/Z) \sim Dirichlet(\alpha_1 + \alpha_2, \alpha_3, ..., \alpha_K)$. ■

Exercise 5.16 (Decimation property) *Let* $(\pi_1, ..., \pi_K) \sim Dirichlet(\alpha_1, ..., \alpha_K)$, *and let* $\tau \sim Beta(\beta)$. *Show that* $\tau\pi_1, (1 - \tau)\pi_1, \pi_2, \ldots, \pi_K) \sim Dirichlet(\beta\alpha_1, (1 - \beta)\alpha_1, \alpha_2, \ldots, \alpha_K)$.

Exercise 5.17 *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.*

Proof: Since partition $\{P_1, ..., P_n\}$ breaks at multiples of $1/K$, so each P_i contains multiple of intervals $[\frac{k-1}{K}, \frac{k}{K})$. The weights associated with P_i is the sum of weights associated with those intervals inside P_i. Since the distribution over the weights on those intervals is dirichlet, according to last exercise, any combinations of those weights is also dirichlet. ■

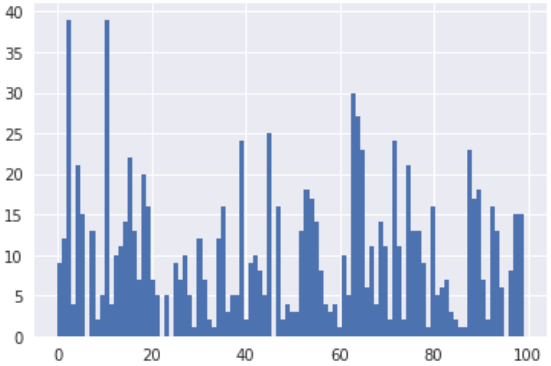
The Dirichlet process extends this idea to arbitrary partitions. Concretely, the Dirichlet process is a distribution over measures[1] on some space $\otimes$, parametrized by some probability distribution H on Ω and some positive scalar α such that for any partition $A_1, \ldots, A_K$ of Ω, the masses 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.

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 $K \to \infty$.

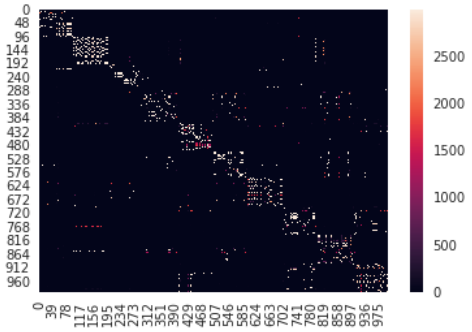
Exercise 5.18 *Return to the MNIST mixture model, and replace your 10-dimensional Dirichlet distribution with a 100-dimensional Dirichlet with parameters $\alpha/100$ for, say, $\alpha = 1$. How many clusters does it use (look at a distribution over multiple samples)? Based on a single sample, what do those clusters look like?*

Proof: The number of images in each cluster is shown in Figure 5.7a. We can see that there are some empty clusters. Also we can see in Figure 5.7b, it shows the co-occurrence matrix and it has less off-diagonal block shown than the one with 10 clusters, which is nice. In Figure 5.7c, we also show the average images within each cluster, and we can find some numbers such as 5, 7, 4, which we cannot find in previous case. ■

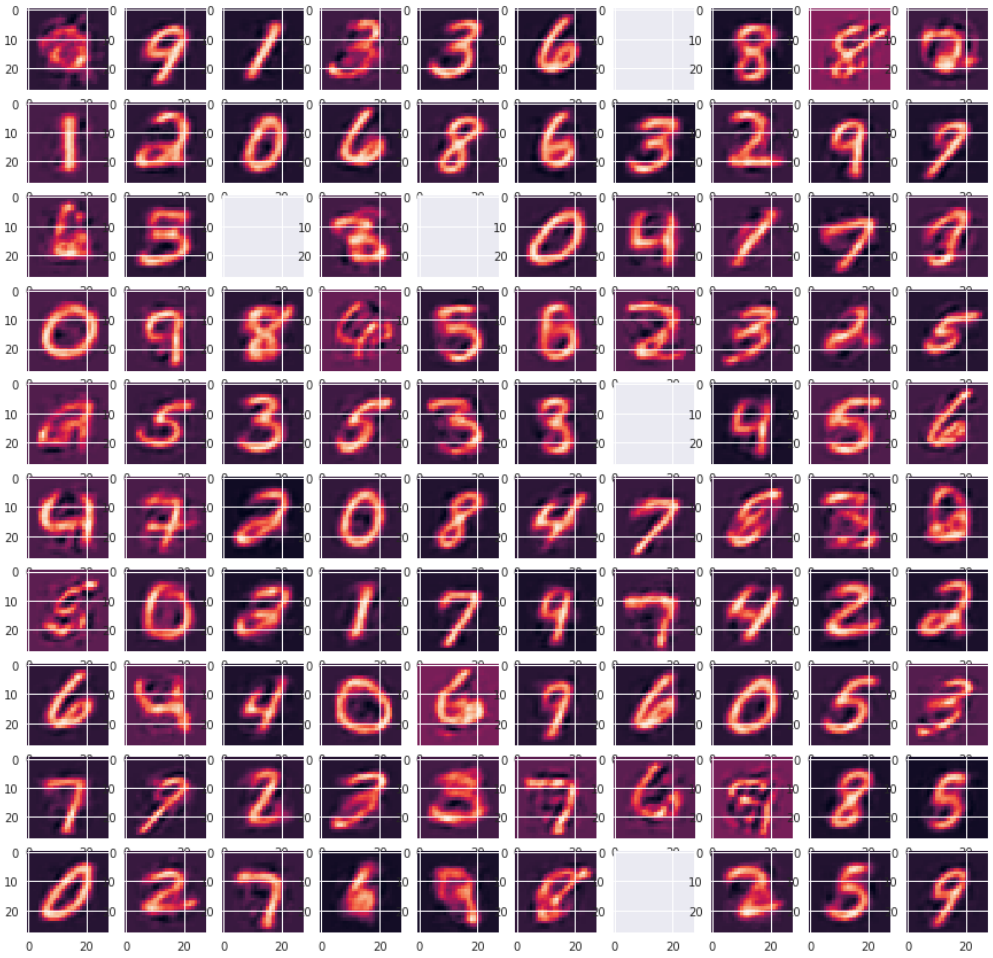
[1] If you're not familiar with measure theory, a measure on some space is just a function that assigns a positive number to every subset of that space. So, a probability is a measure. Area is a measure.



(a) Number of images in each cluster



(b) Heatmap of co-occurrence matrix



(c) Average images for each cluster

Figure 5.7