Sareh Kouchaki

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

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?*

Solution:

From 2.12, we consider where y is a vector of profits; X is a n * d matrix of covariates;

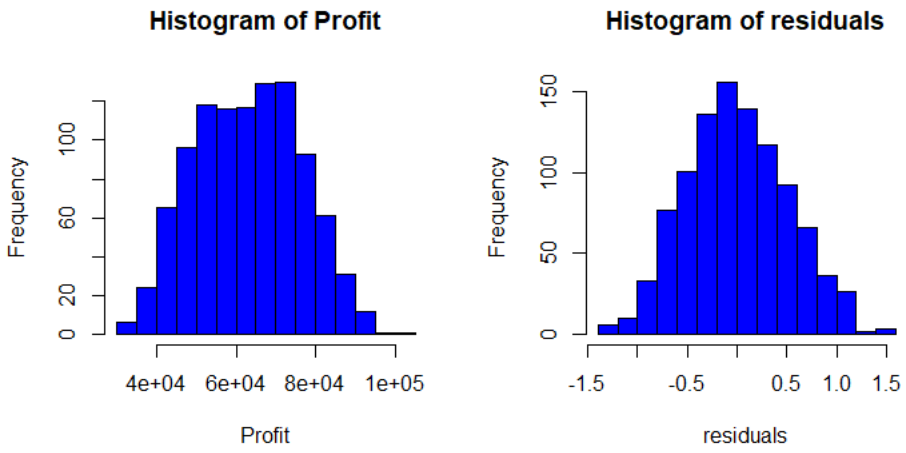
$y|\beta, X \sim \text{Normal}\ (X\beta, (\omega\Lambda)^{-1})$

Priors: $\beta \sim \text{Normal}(\mu, (\omega K)^{-1})$, $\omega \sim \text{Gamma}(a, b)$

Posteriors: $p(\beta|\omega, y) \sim Normal\left(\frac{X^T \Lambda y + k\mu}{K + X^T \Lambda X}, \left(\omega(K + X^T \Lambda X)\right)^{-1}\right)$

and $p(\omega|y) \sim \text{Gamma}\left(a + \frac{n}{2}, b + \frac{1}{2}(\mu^T K \mu + Y^T \Lambda Y - {\mu_n}^T (k + X^T \Lambda X)\mu_n)\right)$

The histograms of residuals and profit are provided in the following. The histogram of residuals shows that the residuals look normal. However, the raw profile data doesn't 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:

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

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.

Solution:

We assume two clusters with different means: μ_1, μ_2 and the same variance σ^2. A latent indicator variable z_i is considered, which assign each observation to one of the two groups.

The prior distribution of the latent variable is assumed to be Bernoulli and therefore the posterior is as following:

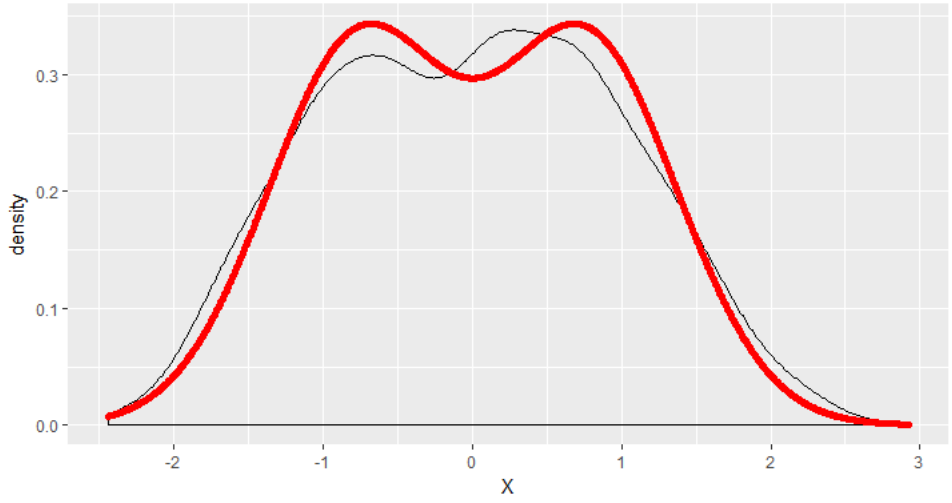
$$\mathrm{p(z|x)} \propto p(z)p(x|z)$$

$$\mathrm{p(z=1|x)} = \frac{p(z=1)p(x|z=1)}{p(z=1)p(x|z=1) + p(z=2)p(x|z=2)} = \frac{0.5p(x|z=1)}{0.5p(x|z=1) + 0.5p(x|z=2)} = \frac{p(x|z=1)}{p(x|z=1) + p(x|z=2)}$$

The means of clusters are considered as normal$(\mu_0, {\sigma_0}^2)$ and the precision is gamma distributed variable. Therefore, the posterior distribution of means and precision are obtained using conjugacy similar to what we did in section2.

Code is available on GitHub. (fix the code)

The plots of $\mathrm{X} = 0.5 * \mathrm{N}(\mu_1, \sigma^2) + 0.5 * \mathrm{N}(\mu_2, \sigma^2)$ along with the plot of raw profit are provided in the following figure.



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?*

Solution:

$$p(Z_i|Z_{\neg i}, X_i, \mu_1, \mu_2, \sigma_1, \sigma_2, \alpha, \beta) = \int p(Z_i|\pi, X_i, \mu_1, \mu_2, \sigma_1, \sigma_2) * P(\pi|Z_{\neg i}, \alpha, \beta)\, d\pi$$

$$\propto \int \pi^{Z_i}(1-\pi)^{1-Z_i} N(X_i|\,\mu_{Z_i}, {\sigma_{Z_i}}^2) * Beta(\alpha, \beta) * \prod_{j \neq i} \pi^{Z_{\neg i}} (1-\pi)^{1-Z_{\neg i}} d\pi$$

$$\propto \int \pi^{Z_i}(1-\pi)^{1-Z_i} N(X_i|\,\mu_{Z_i}, {\sigma_{Z_i}}^2) \pi^{\alpha-1}(1-\pi)^{\beta-1} \pi^{\sum Z_{\neg i}} (1-\pi)^{n-\sum Z_{\neg i}} d\pi$$

$$\propto \int \pi^{\alpha+\sum Z_{\neg i}+Z_i-1}(1-\pi)^{\beta+n-\sum Z_{\neg i}-Z_i} N(X_i|\,\mu_{Z_i}, {\sigma_{Z_i}}^2) d\pi$$

$$\propto N\left(X_i|\mu_{Z_i}, {\sigma_{Z_i}}^2\right) * \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} * \frac{\Gamma(\alpha + \sum Z_{\neg i} + Z_i)\Gamma(\beta + n - \sum Z_{\neg i} - Z_i + 1)}{\Gamma(\alpha + \sum Z_{\neg i} + \beta + n - \sum Z_{\neg i} + 1)}$$

The above distribution is normal $({\mu_{Z_i}, \sigma_{Z_i}}^2)$ $beta - binomial$ $(Z_i|1, \alpha + \sum Z_{\neg i}, \beta + n - \sum Z_{\neg i})$.

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:

$$y \sim \pi_1 * N(\mu_1, \lambda) + \pi_2 * N(\mu_2, \lambda)$$

Where $\lambda = \sigma^2$

$$\mu_1, \mu_2 \sim N(0, \lambda)$$

$$\lambda \sim Gamma(\alpha, \beta)$$

By using the conjugacy, we can find the posterior distribution of parameters, for example:

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

So, we could integrate out the parameters and we will have:

$$p(Z_i|Z_{\neg i}, y, \mu_1, \mu_2, \sigma_1, \sigma_2, \alpha, \beta) = \frac{\Gamma(\alpha+n_1)\Gamma(\beta+n_2)\Gamma(a_1)\Gamma(a_2)}{\Gamma(\alpha+n_1+\beta+n_2){b_1}^{a_1}{b_2}^{a_2}}$$

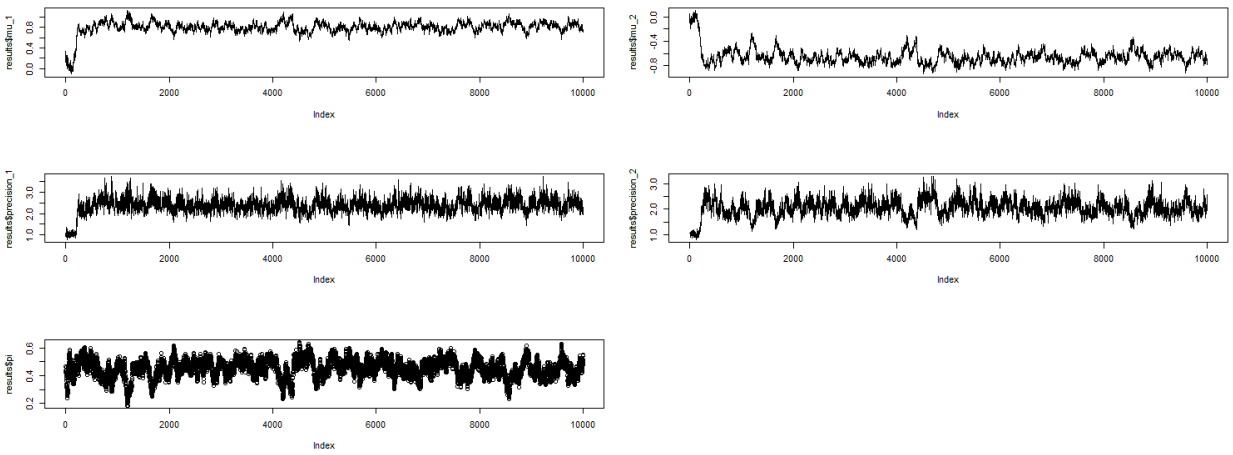
Where $a_1 = \alpha + \frac{n_1}{2}, b_1 = \beta + \frac{\sum_{i:Z_i=1} {y_i}^2}{2} - \frac{(\sum_{i:Z_i=1} y_i)^2}{2n_1}$, $a_2 = \alpha + \frac{n_2}{2}$, $and\ b_2 = \beta + \frac{\sum_{i:Z_i=2} {y_i}^2}{2} - \frac{(\sum_{i:Z_i=2} y_i)^2}{2n_2}$

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.

Solution:

Code is available on GitHub.

Plots of clusters' mean, precision, and pi obtained from 10000 sampling.



Results of Gibbs sampler:

	Mean	Precision
Cluster 1	0.795	2.374
Cluster 2	-0.652	2.041

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

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

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, $E[\pi] = \frac{(\alpha_1,\ldots,\alpha_K)}{\sum_k \alpha_k}$. 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 5.1 below shows the pdf of three Dirichlet distributions represented on the 3-simplex, with samples from those distributions

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

Solution:

(link: https://en.wikipedia.org/wiki/Dirichlet-multinomial_distribution)

The Dirichlet distribution is a conjugate prior for the multinomial distribution. This means that if the prior distribution of the multinomial parameters is Dirichlet then the posterior distribution is also a Dirichlet distribution (with parameters different from those of the prior).

For a random vector of category counts $x = (x_1, x_2, \ldots, x_n)$, distributed according to a multinomial distribution:

$$p(x|p) \sim Multinomial(p_1, p_2, \ldots, p_n)$$

$$(p_1, p_2, \ldots, p_n) \sim Dirichlet\ (\alpha_1, \alpha_2, \ldots, \alpha_n)$$

$$P(x|\alpha) = \int p(x|p)\, P(p|\alpha)\, dp = \frac{(n!)\,\Gamma(\sum \alpha_k)}{\Gamma(n + \sum \alpha_k)} \prod_{k=1}^{n} \frac{\Gamma(x_k + \alpha_k)}{(x_k!)\,\Gamma(\alpha_k)}$$

The obtained distribution is a Dirichlet distribution.

In general, posterior predictive is as following, where D is the observed data, and y is the new datapoint:

$$f(y|D) = \int f(y, \theta|D)\, d\theta = \int f(y|\theta)\, f(\theta|D)\, d\theta$$

**$f(\theta|D) = f(\theta|\alpha) \prod_{y_i \in D} f(y_i|\theta)$

For Dirichlet – Multinomial:

$$p_1, p_2, \ldots, p_k \sim Dir(\alpha_1, \alpha_2, \ldots, \alpha_k)$$

$$y_1, y_2, \ldots, y_k \sim \text{Mult}(p_1, p_2, \ldots, p_k)$$

$$f(p_1, p_2, \ldots, p_k|\alpha_1, \alpha_2, \ldots, \alpha_k) \prod_{y_i \in D} f(y_i|p_1, p_2, \ldots, p_k) \propto \prod_{j=1}^{K} {p_j}^{\alpha_j - 1} \prod_{y_i \in D} \prod_{j=1}^{K} {p_j}^{y_i{}^{(j)}}$$

$$= \prod_{j=1}^{K} {p_j}^{\alpha_j - 1 + \sum_{y_i \in D} y_i{}^{(j)}}$$

This density is a Dirichlet distribution with $\alpha_j + \sum_{y_i \in D} y_i{}^{(j)}$ parameter.

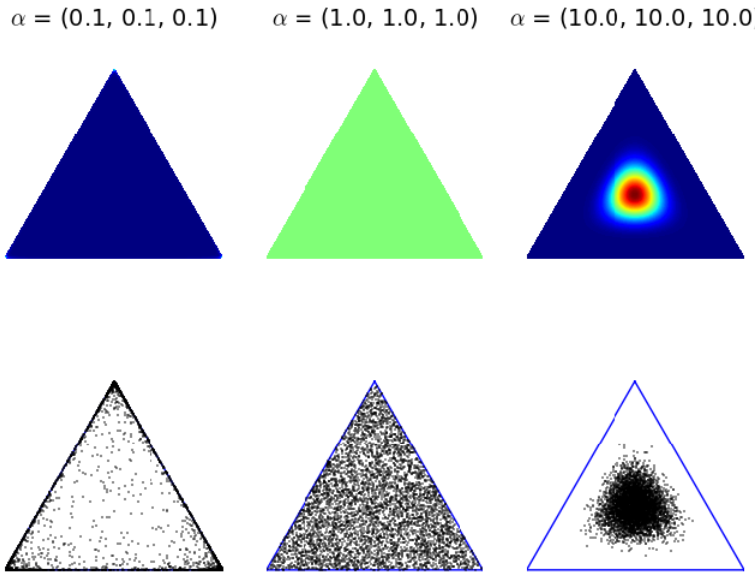


Figure 5.1: PDF and samples from three Dirichlet distributions with parameters α

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:

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.*

Results for $\alpha = (10, 10, \ldots 10)$

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
run1	8	66	87	43	73	33	32	53	37	26
run2	49	65	96	5	2	82	59	37	59	25
run3	86	100	66	68	79	44	2	67	33	59
run4	9	7	1	87	63	63	19	66	91	61
run5	92	10	100	68	90	76	37	94	81	35

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.*

Results for $\alpha = (1, 1, \ldots 1)$

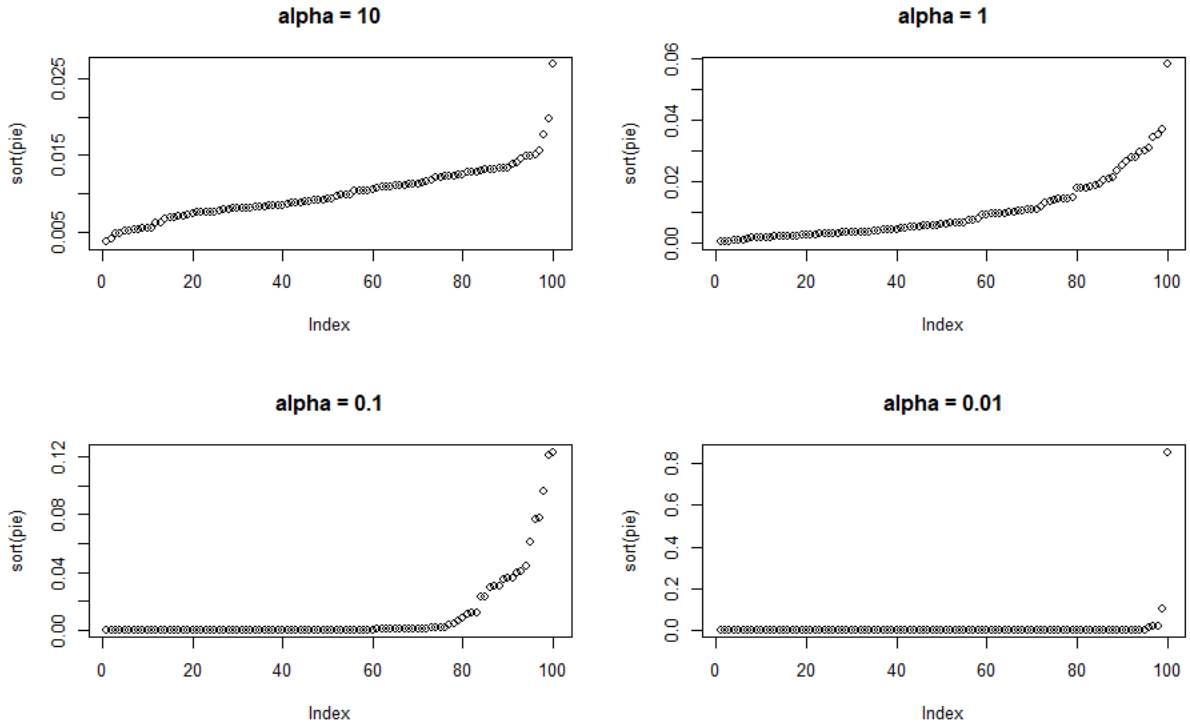
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
run1	40	98	51	65	31	24	73	10	97	72
run2	65	95	80	80	91	1	44	74	65	7
run3	99	53	49	36	14	57	35	3	84	64
run4	18	9	4	82	79	47	86	37	41	86
run5	92	75	43	20	50	9	92	46	20	29

Results for $\alpha = (0.1, 0.1, \ldots 0.1)$

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
run1	15	60	57	62	55	32	15	61	32	74
run2	86	55	85	23	85	16	38	86	86	97
run3	7	35	76	45	79	33	73	45	19	73
run4	85	93	81	91	33	82	33	32	32	85
run5	54	54	54	69	100	10	50	28	50	100

Results for $\alpha = (0.01, 0.01, \ldots 0.01)$

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
run1	54	90	90	90	90	7	90	90	90	90
run2	31	31	31	31	31	31	31	31	31	31
run3	52	52	5	5	5	5	5	5	5	5
run4	63	46	63	63	63	63	63	63	63	45
run5	88	88	88	62	88	88	88	88	88	88


alpha = 10
sort(pie)
0.005
0.015
0.025
0
20
40
60
80
100
Index
alpha = 1
sort(pie)
0.00
0.02
0.04
0.06
0
20
40
60
80
100
Index
alpha = 0.1
sort(pie)
0.00
0.04
0.08
0.12
0
20
40
60
80
100
Index
alpha = 0.01
sort(pie)
0.0
0.2
0.4
0.6
0.8
0
20
40
60
80
100
Index