Technical Report – Department of Mathematical Statistics and Actuarial Science - UFS

Bayesian Fitting of Dirichlet Type I and II Distributions

Sean van der Merwe, Daan de Waal
Department of Mathematical Statistics and Actuarial Science, IB 75, University of the Free State, Box 339, Bloemfontein, 9300, South Africa.

Abstract

The Dirichlet Type I distribution is a multivariate distribution commonly used to model a set of proportions that sum to one. The Dirichlet Type II distribution is a transformation of the Dirichlet Type I distribution and is a multivariate distribution on the positive real numbers with only one more parameter than the number of dimensions. This property implies advantages over common alternatives for small samples. We describe these distributions and show how to fit them using both frequentist and Bayesian methods. The Beta distribution is discussed as a special case. We give an example, report a small simulation study to compare the fitting methods, and discuss the predictive posterior conditional distributions.

Keywords

Dirichlet, Beta, Dirichlet Type II, Bayes, Simulation, Objective prior

1 Introduction

The Dirichlet distribution, the multivariate counterpart to the Beta distribution, is often used to model a set of proportions that sum to one, or less than one. In the frequentist sense the distribution is well-documented and researched - Minka (2012), Ronning (1989), Wicker *et al.* (2008), Narayanan (1992), *etc.* all discuss the various approaches to maximising the likelihood, as well as the method of moments; but Bayesian analysis of this distribution appears to focus entirely on its use as a prior and posterior for the Multinomial distribution. We intend to show that the Dirichlet distribution is useful in its own right and that the Bayesian approach adds to its value.

From Minka (2012) we see that if $X = X_1, \dots, X_{m+1} \sim Dirichlet(k_1, \dots, k_{m+1})$, then the density is

$$f(\underline{x}) = \frac{\Gamma(k_0)}{\prod_{i=1}^{m+1}\Gamma(k_i)}\left(\prod_{i=1}^{m+1} x_i^{k_i-1}\right) \tag{1.1}$$

where $0 < x_i < 1$, $i = 1, \dots, m+1$; $\sum_{i=1}^{m+1} x_i = 1$ and $k_0 = k_1 + \cdots + k_{m+1}$.

We say that this distribution is m dimensional because if any m values of $\underline{x}$ are known then so is the last value.

In the case where $m = 1$ the density (1.1) simplifies to the density of the well-known Beta distribution:

$$f(x_1) = \frac{\Gamma(k_1 + k_2)}{\Gamma(k_1)\Gamma(k_2)} x_1^{k_1 - 1}(1 - x_1)^{k_2 - 1} \tag{1.2}$$

In general, the marginal distributions are: $X_i \sim Beta(k_i, k_0 - k_i)$, $i = 1 \dots m + 1$, with $E(X_i) = \frac{k_i}{k_0}$.

A more interesting property (used to derive the above result on the marginal distributions) is the aggregation property (proved in Frigyik *et al.*, 2010). It says that if two components of a Dirichlet vector are dropped and replaced with their sum then the resulting vector is again Dirichlet with the corresponding parameters replaced in the same way. More explicitly, if $(X_1, \dots, X_K) \sim D(\alpha_1, \dots, \alpha_K)$ then $\big(X_1, \dots, X_i + X_j, \dots, X_K\big) \sim D\big(\alpha_1, \dots, \alpha_i + \alpha_j, \dots, \alpha_K\big)$.

Another way to interpret this property is that the joint marginal distributions are also Dirichlet, in the sense that if $(X_1, \dots, X_m, 1 - \sum_{i=1}^{m} X_i) \sim D(\alpha_1, \dots, \alpha_m, \alpha_{m+1})$, then $\big(X_1, \dots, X_j, 1 - \sum_{i=1}^{j} X_i\big) \sim D\big(\alpha_1, \dots, \alpha_j, \sum_{i=j+1}^{m+1} \alpha_i\big)$. This result becomes very useful later when we discuss the conditional distribution.

The Dirichlet distribution is illustrated in Kotz and Nadarajah (2000, pp. 127-128).

2 Fitting the Dirichlet distribution

Let $\underline{x}_1, \dots, \underline{x}_n$ be an observed sample from the Dirichlet Type I distribution. We combine these vectors as rows of a matrix, so that from here on X_{ij} refers to the j^{th} element of the i^{th} observation. In general, X_j or $j \in \{1, \dots, m + 1\}$ will refer to a specific margin (*e.g.* X_1 refers to the first margin) while $i \in \{1, \dots, n\}$ will refer to an observation.

2.1 Method of moments

Minka (2012) points out that $k_0 = \frac{E(X_1) - E(X_1^2)}{E(X_1^2) - E(X_1)^2}$. By multiplying each expected value by this expression we obtain estimates for each parameter. Explicitly:

$$E\big(X_j\big) = \frac{k_j}{k_0} \Rightarrow k_j = E\big(X_j\big)k_0 = E\big(X_j\big)\left[\frac{E(X_1) - E(X_1^2)}{E(X_1^2) - E(X_1)^2}\right] \tag{2.1}$$

This method does not correspond exactly to the commonly used estimates for the Beta case. An alternative approach is given in Ronning (1989).

The method of moments is used as a starting point for the method of maximum likelihood in this study.

2.2 Method of maximum likelihood

Minka (2012) suggests the following iteration for maximising the likelihood:

$$\psi\big(k_j^{new}\big) = \psi\big(k_0^{old}\big) + \frac{1}{n}\sum_{i=1}^{n} \log X_{ij} \tag{2.2}$$

$\psi(x)$ refers to the digamma function, which occurs throughout this text as both $\frac{d}{dx}\log\Gamma(x)$ and $E[\log G]$ where $G \sim Gamma(x,1)$. We found this method to converge quickly in all cases.

A detailed discussion of the methods available to maximise the likelihood is given in Huang (2008).

2.3 Posterior distribution

Using a Bayesian approach, the parameters of the Dirichlet distribution can be estimated from the posterior distribution (*e.g.* as the mean or mode of the relevant posterior). To begin with an appropriate prior distribution must be specified.

2.3.1 MDI Prior

We consider first the maximal data information (MDI) prior (Zellner, 1971, pp. 41-53). The log of the MDI prior is the expected value of the log density. Since

$$\log f(x) = \log\Gamma(k_0) - \sum_{i=1}^{m+1}\log\Gamma(k_i) + \sum_{i=1}^{m+1}[(k_i-1)\log x_i] \tag{2.3}$$

we have

$$\begin{aligned} E[\log f(X)] &= \log\Gamma(k_0) - \sum_{i=1}^{m+1}\log\Gamma(k_i) + \sum_{i=1}^{m+1}[(k_i-1)E(\log X_i)] \\ &= \log\Gamma(k_0) - \sum_{i=1}^{m+1}\log\Gamma(k_i) + \sum_{i=1}^{m+1}\big[(k_i-1)\big(\psi(k_i)-\psi(k_0)\big)\big]. \end{aligned} \tag{2.4}$$

The log likelihood is

$$\begin{aligned} \ell &= n\log\Gamma(k_0) - n\sum_{i=1}^{m+1}\log\Gamma(k_i) + \sum_{j=1}^{n}\sum_{i=1}^{m+1}\big[(k_i-1)\log x_{ij}\big] \\ &= n\log\Gamma(k_0) - n\sum_{i=1}^{m+1}\log\Gamma(k_i) + \sum_{i=1}^{m+1}\left[(k_i-1)\sum_{j=1}^{n}\log x_{ij}\right] \end{aligned} \tag{2.5}$$

Adding (2.4) and (2.5) then gives the log posterior:

$$\begin{aligned} \log\pi\big(\underline{k}|X\big) &= (n+1)\log\Gamma(k_0) - (n+1)\sum_{i=1}^{m+1}\log\Gamma(k_i) \\ &\quad + \sum_{i=1}^{m+1}\left[(k_i-1)\left[\big(\psi(k_i)-\psi(k_0)\big) + \sum_{j=1}^{n}\log x_{ij}\right]\right] + c \end{aligned} \tag{2.6}$$

where c is an unknown constant.

Wicker *et al.* (2008) explain that the likelihood is globally concave. The addition of the prior does not appear to change this property, simplifying the process of finding the mode greatly, as the mode is simply the set of parameter values where the posterior reaches its peak. We employed a simple gradient ascent algorithm which only requires the first derivatives:

$$\frac{\partial \log \pi}{\partial k_i} = (n+1)[\psi(k_0) - \psi(k_i)] + \sum_{j=1}^{n} \log x_{ij} + \big(\psi(k_i) - \psi(k_0)\big)$$
$$+ (k_i - 1)\psi'(k_i) - \sum_{j=1}^{m+1} \big(k_j - 1\big)\psi'(k_0) \qquad (2.7)$$
$$= n[\psi(k_0) - \psi(k_i)] + \sum_{j=1}^{n} \log x_{ij} + (k_i - 1)\psi'(k_i) - \psi'(k_0)(k_0 - m - 1)$$

Alternatively, we can simulate a sample from the posterior (2.6) and calculate the mean, which can be done using the Metropolis-Hastings (MH) algorithm. See Robert and Casella (2004, pp. 267-301) for an in-depth general discussion of this algorithm.

The MH algorithm requires the specification of a suitable jump distribution for choosing candidate values. We chose to use the Multivariate Normal distribution, *i.e.* $\underline{k}^c \sim N_{m+1}\big(\underline{k}^i, \Sigma\big)$ where Σ is a diagonal matrix of constants that affect the acceptance rate and required burn-in period. We chose these constants by crudely matching the peak of the posterior to the peak of a Multivariate Normal distribution, one marginal at a time using the method of percentiles (finding the percentiles using stepwise linear interpolation).

In this way the algorithm is simplified to:

Accept candidate $\underline{k}^c$ as a new observation $\big(\underline{k}^{i+1}\big)$ from the posterior if and only if

$\log \pi\big(\underline{k}^c|X\big) - \log \pi\big(\underline{k}^i|X\big) > \log u$ where u is drawn randomly from the standard Uniform distribution.

It is worth noting that the above method works very well for any finite number of dimensions, including the Beta case $(m = 1)$.

2.3.2 Jeffreys' Prior

An alternative to the MDI prior is the Jeffreys prior (Jeffreys, 1961) which is derived as follows:

Let $g = \log f(x) = \log \Gamma(k_0) - \sum_{i=1}^{m+1} \log \Gamma(k_i) + \sum_{i=1}^{m+1} [(k_i - 1) \log x_i]$. Then

$$\frac{\partial g}{\partial k_i} = \psi(k_0) - \psi(k_i) + \log x_i,$$
$$\frac{\partial^2 g}{\partial k_i^2} = \psi'(k_0) - \psi'(k_i),$$
$$\frac{\partial^2 g}{\partial k_i \partial k_j} = \psi'(k_0), \qquad (2.8)$$
$$-E\left[\frac{\partial^2 g}{\partial k_i^2}\right] = \psi'(k_i) - \psi'(k_0),$$
$$-E\left[\frac{\partial^2 g}{\partial k_i \partial k_j}\right] = -\psi'(k_0)$$

and thus

$$\text{Jeffreys Prior} \propto \begin{vmatrix} \psi'(k_1)-\psi'(k_0) & -\psi'(k_0) & \cdots & -\psi'(k_0) \\ -\psi'(k_0) & \psi'(k_2)-\psi'(k_0) & & \vdots \\ \vdots & & \ddots & \vdots \\ -\psi'(k_0) & \cdots & \cdots & \psi'(k_{m+1})-\psi'(k_0) \end{vmatrix}^{\frac{1}{2}} \tag{2.9}$$

In the Beta case $(m = 1)$ we get

$$\sqrt{\psi'(k_1)\psi'(k_2)-\psi'(k_0)[\psi'(k_1)+\psi'(k_2)]}. \tag{2.10}$$

When $m = 2$ we get:

$$\{\psi'(k_1)\psi'(k_2)\psi'(k_3)-\psi'(k_0)[\psi'(k_1)\psi'(k_2)+\psi'(k_1)\psi'(k_3)+\psi'(k_2)\psi'(k_3)]\}^{\frac{1}{2}} \tag{2.11}$$

In general the Jeffreys prior is proportional to the square root of

$$\begin{aligned} &\prod_{l=1}^{m+1}\psi'(k_l)-\psi'(k_0)\sum_{l=1}^{m+1}\prod_{\substack{j=1\\ j\neq l}}^{m+1}\psi'(k_j) \\ &=\prod_{l=1}^{m+1}\psi'(k_l)-\psi'(k_0)\sum_{l=1}^{m+1}\frac{1}{\psi'(k_l)}\prod_{j=1}^{m+1}\psi'(k_j) \\ &=\prod_{l=1}^{m+1}\psi'(k_l)\left[1-\psi'(k_0)\sum_{l=1}^{m+1}\frac{1}{\psi'(k_l)}\right] \end{aligned} \tag{2.12}$$

Thus the log Jeffreys prior is given by

$$\begin{aligned} &0.5\log\prod_{l=1}^{m+1}\psi'(k_l)\left[1-\psi'(k_0)\sum_{l=1}^{m+1}\frac{1}{\psi'(k_l)}\right] \\ &=0.5\sum_{l=1}^{m+1}\log\psi'(k_l)+0.5\log\left[1-\psi'(k_0)\sum_{l=1}^{m+1}\frac{1}{\psi'(k_l)}\right] \end{aligned} \tag{2.13}$$

so that the log posterior is equal to

$$\begin{aligned} 0.5\sum_{l=1}^{m+1}\log\psi'(k_l)+0.5\log\left[1-\psi'(k_0)\sum_{l=1}^{m+1}\frac{1}{\psi'(k_l)}\right]+n\log\Gamma(k_0)-n\sum_{i=1}^{m+1}\log\Gamma(k_i) \\ +\sum_{i=1}^{m+1}\left[(k_i-1)\sum_{j=1}^{n}\log x_{ij}\right]+c. \end{aligned} \tag{2.14}$$

The first derivatives (needed to find the mode) are found to be:

$$\begin{aligned} \frac{\partial\log\pi}{\partial k_i} &= n[\psi(k_0)-\psi(k_i)]+\sum_{j=1}^{n}\log x_{ij}+0.5\frac{\psi''(k_i)}{\psi'(k_i)} \\ &\quad -0.5\left[\psi''(k_0)\sum_{l=1}^{m+1}\frac{1}{\psi'(k_l)}-\psi'(k_0)\psi''(k_i)\big(\psi'(k_i)\big)^{-2}\right] \\ &\quad *\left[1-\psi'(k_0)\sum_{l=1}^{m+1}\frac{1}{\psi'(k_l)}\right]^{-1} \end{aligned} \tag{2.15}$$

The method of simulation is the same as for the MDI prior.

For an overview of the derivation of objective priors and examples see Yang and Berger (1998).

3 The Dirichlet Type II Distribution

Consider again $X(m,1) \sim Dirichlet(k_1, \dots, k_{m+1})$. If we transform this distribution using the formula

$$Y_i = \frac{X_i}{X_{m+1}}, \qquad i = 1, \dots, m \text{ and } X_{m+1} = 1 - \sum_{j=1}^{m} X_j \tag{3.1}$$

then $Y(m,1) \sim Dirichlet\ Type\ \text{II}(k_1, \dots, k_{m+1})$.

The density is now

$$f\left(\underline{y}\right) = \frac{\Gamma(k_0)}{\prod_{i=1}^{m+1}\Gamma(k_i)}\left(\prod_{i=1}^{m} y_i^{k_i-1}\right)(1 + y_1 + \cdots + y_m)^{-k_0} \text{ where } Y_i > 0, \quad i = 1, \dots, m. \tag{3.2}$$

and the log density:

$$\log f\left(\underline{y}\right) = \log\Gamma(k_0) - \sum_{i=1}^{m+1}\log\Gamma(k_i) - k_0 \log\left(1 + \sum_{i=1}^{m} y_i\right) + \sum_{i=1}^{m}[(k_i - 1)\log y_i] \tag{3.3}$$

Thus, given a sample $\left(\underline{y}_1, \dots, \underline{y}_n\right)$, the log likelihood is

$$n\log\Gamma(k_0) - n\sum_{i=1}^{m+1}\log\Gamma(k_i) - k_0\sum_{j=1}^{n}\log\left(1 + \sum_{i=1}^{m} y_{ij}\right) + \sum_{j=1}^{n}\sum_{i=1}^{m}\left[(k_i - 1)\log y_{ij}\right]. \tag{3.4}$$

Again we consider the MDI prior. The log prior is:

$$\begin{aligned} E\left[\log f\left(\underline{Y}\right)\right] = \log\Gamma(k_0) - \sum_{i=1}^{m+1}\log\Gamma(k_i) - k_0 E\left[\log\left(1 + \sum_{i=1}^{m} Y_i\right)\right] \\ + \sum_{i=1}^{m}\left[(k_i - 1)E[\log Y_i]\right] \end{aligned} \tag{3.5}$$

where

$$E[\log Y_i] = E[\log\Gamma(k_i, 1) - \log\Gamma(k_0 - k_i, 1)] = \psi(k_i) - \psi(k_0 - k_i) \tag{3.6}$$

and

$$E\left[\log\left(1 + \sum_{i=1}^{m} Y_i\right)\right] = -E\left[\log\left(1 - \sum_{i=1}^{m} X_i\right)\right] = -E[\log X_{m+1}] = \psi(k_0) - \psi(k_{m+1}). \tag{3.7}$$

The log prior thus simplifies to

$$\log\Gamma(k_0) - \sum_{i=1}^{m+1} \log\Gamma(k_i) - k_0[\psi(k_0) - \psi(k_{m+1})] + \sum_{i=1}^{m} (k_i - 1)[\psi(k_i) - \psi(k_0 - k_i)] \tag{3.8}$$

and the log posterior to

$$(n+1)\left[\log\Gamma(k_0) - \sum_{i=1}^{m+1} \log\Gamma(k_i)\right] - k_0\left\{[\psi(k_0) - \psi(k_{m+1})] + \sum_{j=1}^{n} \log\left(1 + \sum_{i=1}^{m} y_{ij}\right)\right\} + \sum_{i=1}^{m} (k_i - 1)[\psi(k_i) - \psi(k_0 - k_i)] + \sum_{j=1}^{n}\sum_{i=1}^{m}\left[(k_i - 1)\log y_{ij}\right]. \tag{3.9}$$

While it is possible to work directly with the above distribution, including simulating from the posterior as before, we prefer to transform a sample from the Dirichlet Type II distribution to a sample from the Dirichlet Type I distribution and apply the methods discussed in Section 2. The transformation is the inverse of the previous transformation (3.1), *i.e.*

$$X_i = \frac{Y_i}{1 + \sum_{j=1}^{m} Y_j}, \qquad i = 1, \ldots, m. \tag{3.10}$$

This approach is illustrated in Section 4.

4 Example

Fisher's Iris flower data is a well-known dataset with $m = 4$ variables and three populations of size 50 each. It contains observations on the four variables $Y_1 =$ sepal length, $Y_2 =$ sepal width, $Y_3 =$ petal length and $Y_4 =$ petal width of the species Setosa, Versicolor and Virginica.

We first assume that the distribution of a set of measurements

$$(X_{1j}X_{2j}, X_{3j}, X_{4j}) = \left(\frac{Y_{1j}}{1 + \sum_{i=1}^{4} Y_{ij}}, \frac{Y_{2j}}{1 + \sum_{i=1}^{4} Y_{ij}}, \frac{Y_{3j}}{1 + \sum_{i=1}^{4} Y_{ij}}, \frac{Y_{4j}}{1 + \sum_{i=1}^{4} Y_{ij}}\right) \tag{4.1}$$

can be described through a Dirichlet distribution $D(\alpha_1, .., \alpha_5)$ for each population, say $X^{(i)} \sim D_4\left\{\underline{\alpha}^{(i)} = \left(\alpha_j^{(i)}, j = 1, .., 5\right)\right\}, i = 1,2,3.$

We proceed to fit Dirichlet distributions on the four variables for each species by transforming as above. We begin by applying the method of moments to obtain initial parameter estimates, as this method is not iterative. We then apply the gradient ascent approach to the posterior distribution using the MDI prior, and arrive at the following estimates of the Dirichlet parameters:

$$\begin{aligned} \hat{\alpha}^{(SET)} &= \{604.71, 412.57, 175.49, 27.78, 120.02\} \\ \hat{\alpha}^{(VER)} &= \{378.31, 176.39, 270.87, 84.17, 64.15\} \\ \hat{\alpha}^{(VIR)} &= \{427.61, 193.26, 360.36, 131.62, 65.37\}. \end{aligned} \tag{4.2}$$

A basic test of goodness of fit is to simulate replicate samples and compare them to the original data visually (see for example Gelman *et al.*, 2004, pp. 159 to 171). Doing this with the iris data produces graphs such as Figure 4.1. There appears to be sufficient correspondence between the observed data (dark colours) and the simulated data (lighter colours). The correspondence is in all four dimensions (the sizes of the spheres indicate the fourth dimension).

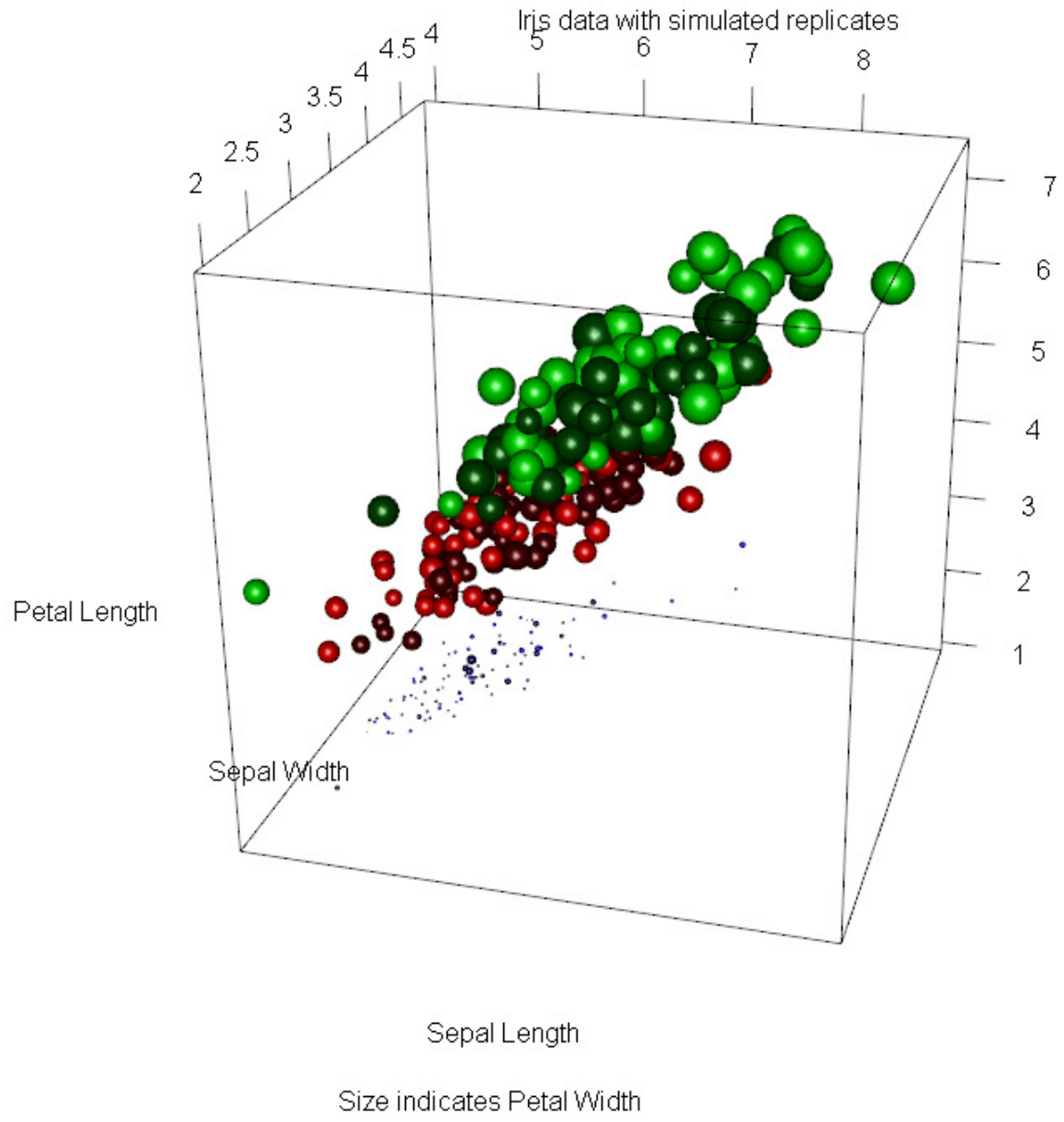

Figure 4.1: Iris data with simulated replicate samples on Dirichlet Type 2 scale. Dark colours indicate the original data and lighter colours the simulated replicate. Green is Virginica, red is Versicolor and blue is Setosa.

All methods were implemented in the statistical package R (R Core Team, 2012). Packages used include the 'parallel' package for increasing the speed of the simulation study in Section 5 (R Core Team, 2012) and the 'rgl' package for drawing the graphs in Figures 4.1 and 6.1 (Adler & Murdoch, 2012).

5 Simulation Study

We wish to compare the accuracy of the different approaches to estimating the parameters of a Dirichlet distribution. To this end we simulate samples from this distribution for a variety of combinations of sample size, dimension and parameter values. For each combination we simulate 5000 samples. For each sample we estimate the parameters in 6 ways and compare the estimates to the true values. We summarise the discrepancies using the root mean square percentage error (RMSPE) in order to attempt to compare accuracy across varying parameter values.

Simulation of a value from the Dirichlet distribution can be done in at least three ways. The simplest method (used in this study) is to consider a Dirichlet observation as the relative sizes of Gamma draws, *i.e.* draw $y_i \sim Gamma(k_i, 1)$ for $i = 1, \dots, m+1$ and then set $x_i = \frac{y_i}{\sum_{j=1}^{m+1} y_j}$. This gives a draw $\underline{x} \sim D(k_1, \dots, k_{m+1})$.

A second method relies on the fact that the univariate marginal and conditional distributions are Beta and simulates values one dimension at a time conditional on the previous dimensions. A third method by Matsui *et al.* (2010) uses a path coupling approach.

The results of the simulation study are given in Table 5.1 below:

No.	RMSPE	Mode - Jeffreys	ML	Mean - Jeffreys	Mode - MDI	Mean - MDI	MOM
1	**n=25; k=(3,3,3)**	21.9%	25.2%	25.9%	27.2%	34.1%	38.6%
2	**n=25; k=(3,3,3,3,3)**	16.8%	19.0%	19.7%	20.9%	26.0%	43.0%
3	**n=50; k=(3,3,3)**	15.1%	16.3%	16.6%	17.0%	19.8%	23.9%
4	**n=50; k=(3,3,3,3,3)**	11.7%	12.4%	12.7%	13.0%	15.0%	25.7%
5	**n=90; k=(3,3,3)**	11.1%	11.6%	11.8%	11.9%	13.2%	17.1%
6	**n=90; k=(3,3,3,3,3)**	8.7%	9.1%	9.2%	9.3%	10.2%	18.3%
7	**n=120; k=(3,3,3,3,3,3,3,3)**	6.5%	6.7%	6.8%	6.9%	7.4%	16.3%
8	**n=25; k=(4,4,4)**	21.2%	24.5%	25.3%	26.7%	34.2%	38.2%
9	**n=25; k=(100,100,100)**	35.3%	31.5%	30.1%	36.0%	34.9%	38.9%
10	**n=25; k=(4,20,100)**	31.6%	27.4%	25.7%	33.6%	34.2%	41.6%
11	**n=50; k=(4,4,4)**	15.1%	16.3%	16.8%	17.1%	20.0%	24.0%
12	**n=50; k=(100,100,100)**	19.3%	19.2%	18.9%	19.9%	21.0%	23.6%
13	**n=50; k=(4,20,100)**	16.7%	17.2%	17.1%	18.1%	20.0%	26.1%
14	**n=90; k=(4,4,4)**	10.9%	11.4%	11.6%	11.7%	12.9%	16.4%
15	**n=90; k=(100,100,100)**	12.0%	13.2%	13.1%	12.4%	14.2%	16.4%
16	**n=90; k=(4,20,100)**	11.1%	12.0%	12.1%	11.8%	13.4%	18.4%

Table 5.1: Performance of parameter estimation methods as the sample size, parameter magnitude and number of dimensions are varied. The effect of sample size can be seen by moving from row 1 to 7 or 8 to 16. The effect of parameter magnitude is evident in rows 8 vs 9 vs 10, 11 vs 12 vs 13 or 14 vs 15 vs 16. The effect of the number of dimensions can be seen by comparing rows 1 vs 2, 3 vs 4 or 5 vs 6 vs 7.

There are many things worth noting in Table 5.1. The first is that in almost all cases the errors increase as we move from left to right in the table. This means that, in general, the posterior mode using the Jeffreys prior appears to perform the best, followed by the maximum likelihood estimator.

The second is that as some or all parameter values become large then the mean of the posterior simulations (where the mean is taken on each dimension individually) performs better. Thus, if another method is applied and the suggested parameter values are large then this method is recommended.

The third is that the accuracy of all estimates appears to be heavily dependent on sample size. Increased sample size can be expected to yield increased accuracy but the extent to which it does so is remarkable.

The fourth is that increasing the dimension appears to increase the accuracy of the estimates.

We also specifically investigated the performance of these methods for the Beta distribution. In this case we vary only the sample size and parameter values. The results are given in Table 5.2 below:

No.	RMSPE	Mode - Jeffreys	ML	Mean - Jeffreys	Mode - MDI	Mean - MDI	MOM
1	**n=60; k=(0.6,0.6)**	17.5%	18.9%	N/A	19.5%	N/A	21.9%
2	**n=60; k=(6,6)**	18.1%	19.5%	20.4%	20.1%	23.1%	19.6%
3	**n=60; k=(60,60)**	21.1%	21.9%	24.1%	22.4%	26.7%	21.9%
4	**n=60; k=(600,600)**	20.7%	20.7%	22.3%	20.7%	23.1%	20.7%
5	**n=60; k=(6,60)**	19.8%	21.3%	22.8%	21.8%	26.0%	22.6%
6	**n=25; k=(3,3)**	30.5%	36.6%	38.3%	39.0%	50.0%	36.6%
7	**n=50; k=(3,3)**	20.8%	23.0%	23.3%	23.8%	28.4%	23.1%
8	**n=75; k=(3,3)**	16.1%	17.1%	17.5%	17.6%	20.1%	17.2%
9	**n=100; k=(3,3)**	14.3%	15.0%	15.2%	15.3%	17.0%	15.3%

Table 5.2: Performance of parameter estimation methods as the sample size (rows 6 to 9) and parameter magnitude (rows 1 to 5) are varied in the case of the Beta distribution.

The patterns are the same as was observed in Table 5.1 - in all rows we see that the posterior mode using the Jeffreys prior produced the most accurate estimates. What is different in this table is the inclusion of the case where the parameters are all very small (less than 1). In this situation the MH method described in Section 2 requires modifications in order to produce useful simulations. While we were able to make these modifications, we did not include the results here as they are technically from a different algorithm.

6 Predictive posterior conditional distribution

We consider the situation where a number of variables are jointly distributed Dirichlet and we wish to predict the values of a subset of variables, given values for the remainder of the variables. We can do this using the predictive posterior conditional distribution.

Initially assume that the parameter values are known, *i.e.* $(X_1, \dots, X_m)|\underline{k} \sim D(k_1, \dots, k_m, k_{m+1})$ and we are interested in the distribution of $(X_1, \dots, X_j|X_{j+1}, \dots, X_m)$. We note that

$$
\begin{aligned}
f\left(x_1,\dots,x_j\middle|X_{j+1}=x_{j+1},\dots,X_m=x_m\right) &= \frac{f(x_1,\dots,x_m)}{f\left(x_{j+1},\dots,x_m\right)} \\
&= \frac{\frac{\Gamma\left(\sum_{i=1}^{m+1}k_i\right)}{\prod_{i=1}^{m+1}\Gamma(k_i)}\left(\prod_{i=1}^{m}x_i^{k_i-1}\right)\left(1-\sum_{i=1}^{m}x_i\right)^{k_{m+1}-1}}{\frac{\Gamma\left(\sum_{i=1}^{m+1}k_i\right)}{\prod_{i=\mathrm{j}+1}^{m}\Gamma(k_i)\,\Gamma\left(k_{m+1}+\sum_{i=1}^{j}k_i\right)}\left(\prod_{i=\mathrm{j}+1}^{m}x_i^{k_i-1}\right)\left(1-\sum_{i=j+1}^{m}x_i\right)^{k_{m+1}+\sum_{i=1}^{j}k_i-1}} \\
&= \frac{\Gamma\left(k_{m+1}+\sum_{i=1}^{j}k_i\right)}{\Gamma(k_{m+1})\prod_{i=1}^{j}\Gamma(k_i)}\left(\prod_{i=1}^{j}x_i^{k_i-1}\right)\left(1-\sum_{i=1}^{m}x_i\right)^{k_{m+1}-1}\left(1-\sum_{i=j+1}^{m}x_i\right)^{-\left(k_{m+1}+\sum_{i=1}^{j}k_i-1\right)}
\end{aligned}
\tag{6.1}
$$

If we set $\delta=\left(1-\sum_{i=j+1}^{m}x_i\right)^{-1}$, which is a constant in this situation, and then set $w_i=\delta x_i$, $i=1,\dots,j$ then $f\left(w_1,\dots,w_j\middle|X_{j+1}=x_{j+1},\dots,X_m=x_m\right)$ becomes

$$
\frac{\Gamma\left(k_{m+1}+\sum_{i=1}^{j}k_i\right)}{\Gamma(k_{m+1})\prod_{i=1}^{j}\Gamma(k_i)}\left(\prod_{i=1}^{j}w_i^{k_i-1}\right)\left(1-\sum_{i=1}^{j}w_i\right)^{k_{m+1}-1}. \tag{6.2}
$$

From this expression we see that $\left(W_1,\dots,W_j\right)|X_{j+1}=x_{j+1},\dots,X_m=x_m\sim D\left(k_1,\dots,k_j,k_{m+1}\right)$.

Since $\frac{W_i}{1-\sum_{l=1}^{j}W_l}=\frac{\delta X_i}{1-\delta\sum_{l=1}^{j}X_l}=\frac{X_i}{\frac{1}{\delta}-\sum_{l=1}^{j}X_l}=\frac{X_i}{1-\sum_{l=1}^{m}X_l}=Y_i$, we also note that $\left(Y_1,\dots,Y_j\right)|X_{j+1}=x_{j+1},\dots,X_m=x_m\sim D2\left(k_1,\dots,k_j,k_{m+1}\right)$.

Thus, given a set of parameters and a set of known values for some components we can easily calculate quantities of interest. For example, we can simulate from the conditional distribution to calculate prediction intervals.

In practise the parameters are unknown and must be estimated from data. This carries uncertainty across to the predictions, uncertainty that must be taken into account. We adjust for the uncertainty using the predictive posterior conditional distribution.

Practically, the process works as follows:

1. Given a data matrix of size n by $(m+1)$, we simulate a large number of sets of parameters from the posterior distribution. Suppose we simulate T sets and store them in a T by $(m+1)$ parameter matrix K.
2. For each set of simulated parameters we simulate a set of $w's$ from a $D\left(k_{t,1},\dots,k_{t,j},k_{t,m+1}\right)$, $t=1,\dots,T$.
3. Divide each set of $w's$ by δ to obtain predictions for the unknown x values.
4. Combine the predictions to calculate values of interest, since these simulations constitute an approximation of the predictive posterior conditional distribution.

In the case where the given values are on the Dirichlet Type II scale we follow a different approach where we apply the aggregation property directly:

We know that $(X_1, \dots, X_j) \sim D\left(k_1, \dots, k_j, \sum_{i=j+1}^{m+1} k_i\right)$ and that $X_i = \frac{Y_i}{1+\sum_{l=1}^{m} Y_l}$. Thus, given $(Y_{j+1}, \dots, Y_m)$ we have that

$$\frac{\frac{(Y_1, \dots, Y_j)}{1+\sum_{l=1}^{m} Y_l}}{1-\sum_{i=1}^{j}\left(\frac{Y_i}{1+\sum_{l=1}^{m} Y_l}\right)} \sim D2\left(k_1, \dots, k_j, \sum_{i=j+1}^{m+1} k_i\right). \tag{6.3}$$

Luckily, this expression simplifies to

$$\frac{(Y_1, \dots, Y_j)}{1+\sum_{l=j+1}^{m} Y_l}, \tag{6.4}$$

so, if we define $\beta = \left(1+\sum_{l=j+1}^{m} Y_l\right)$ and $V_i = \frac{Y_i}{\beta}$ then

$$(V_1, \dots, V_j)|Y_{j+1} = y_{j+1}, \dots, Y_m = y_m \sim D2\left(k_1, \dots, k_j, \sum_{i=j+1}^{m+1} k_i\right). \tag{6.5}$$

We can also define $U_i = \frac{V_i}{1+\sum_{l=1}^{j} V_l}$ and note that

$$(U_1, \dots, U_j)|Y_{j+1} = y_{j+1}, \dots, Y_m = y_m \sim D\left(k_1, \dots, k_j, \sum_{i=j+1}^{m+1} k_i\right). \tag{6.6}$$

We can use either of these and appropriately modify steps 2 and 3 in the given process to calculate desired quantities. For example, if we simulate a set of $u's$ then we note that $y_i = \frac{\beta u_i}{1-\sum_{l=1}^{j} u_l}$.

As an example, let us again consider the Iris data from Section 4. Suppose we observe a new flower from the Versicolor family with a sepal length of 6 and a sepal width of 3, but we are unable to measure the petals. We apply the above process with $T = 9999$ and find that the predictive posterior conditional distribution has the following form:

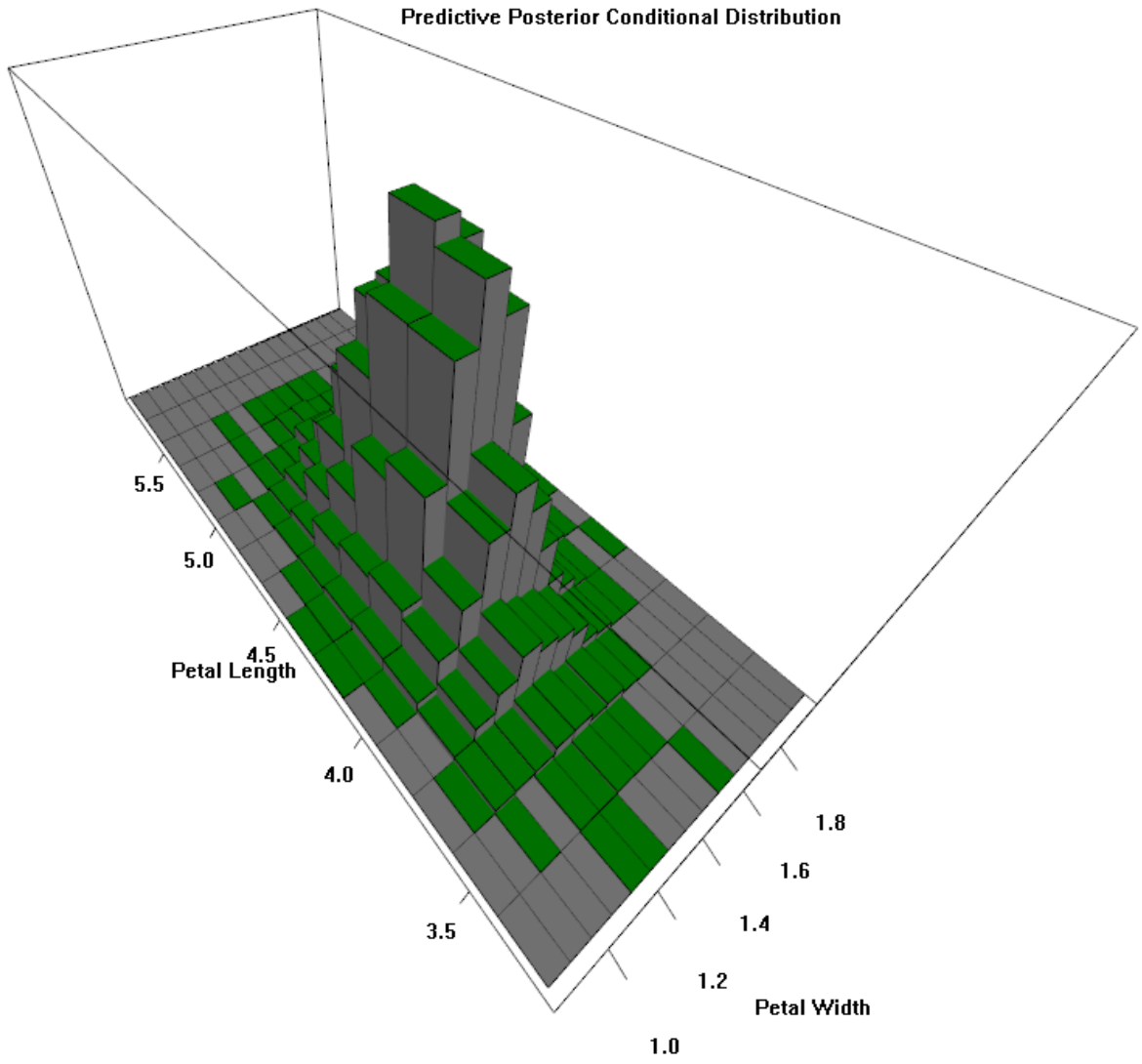

Figure 6.1: Bivariate histogram to illustrate the predictive posterior conditional distribution of the petal size given that the sepal length and width are 6 and 3 respectively for the Versicolor species.

By taking means we predict a petal length of 4.38 and a petal width of 1.36, which seems reasonable looking at Figure 4.1.

7 Multiple Imputation of Dirichlet Type II Data

As an additional example we include an application of the methods discussed. We consider a set of monthly total rainfall figures collected at five stations around the city of Bloemfontein in central South Africa. We only consider months where measurable amounts of rain fell. We have 37 months where all five stations produced measurements and 58 months where a station's data is missing and needs to be imputed.

For an overview of multiple imputation see Schafer & Olsen (1998).

We assume that the rainfall measurements follow a Dirichlet Type II distribution with unknown parameters. We proceed through the following steps:

1. Estimate the six parameters of the distribution using only the 37 complete months.
2. Pick random values for all the missing measurements from the respective predictive posterior conditional distribution of each. This is done one hole at a time since each distribution is different.
3. Estimate the six parameters of the distribution using all 95 months.
4. Pick random values for all the missing measurements from the respective predictive posterior conditional distribution of each.
5. Repeat steps 3 and 4 a number of times, preferably until the parameter estimates converge in distribution.
6. Store the last set of completed data and last set of parameter estimates.
7. Repeat steps 1 to 6 a number of times (min. 5) to obtain a collection of imputed data sets.

The parameter estimates that are produced as a result of this approach are given in Table 7.1. It is clear that parameters k_2 and k_4 would be underestimated if only the complete data were used.

Parameter estimates	**k1**	**k2**	**k3**	**k4**	**k5**	**k6**
Complete data only	3.57	3.49	2.72	2.80	3.62	0.42
Imputation 1	3.43	3.83	2.78	3.13	3.97	0.43
Imputation 2	3.43	3.89	2.65	2.91	4.00	0.43
Imputation 3	3.50	3.82	2.77	3.03	4.03	0.43
Imputation 4	3.46	3.83	2.75	3.12	4.07	0.43
Imputation 5	3.55	3.94	2.80	3.17	4.05	0.43
Imputation 6	3.46	3.81	2.78	3.15	3.98	0.43
Imputation 7	3.63	4.04	2.84	3.33	4.17	0.44
Imputation 8	3.20	3.49	2.54	2.83	3.70	0.42
Imputation 9	3.28	3.65	2.65	2.93	3.66	0.42
Imputation 10	3.59	4.02	2.85	2.98	4.12	0.44

Table 7.1: Parameter estimates following multiple imputation.

An extract of imputed data values is given in Table 7.2 for illustration.

Month	**Station 1**	**Station 2**	**Station 3**	**Station 4**	**Station 5**
2003m4	5.8	6.5	8.5	**6.4**	9.4
2003m8	6.3	3.4	2.1	**2.6**	4.6
2003m9	30.4	18	20.2	**28.3**	20.3
2003m11	47.6	49.7	16.3	**55.1**	42.8
2003m12	24.3	52.4	0.7	**7.9**	45.2
2004m1	5.6	60.6	3.2	**2.4**	49.2
2004m3	73.8	85.3	**75.7**	46.7	151.5

Table 7.2: Extract from final round of Imputation 2 showing typical imputed values.

It is clear that this approach provides a valid alternative to the commonly used Multivariate Normal imputation (see von Hippel, 2012 for an explanation of why this is important).

8 Conclusion

We have found that the Dirichlet distribution family is a useful distribution when modelling random variables that take on positive values. We see that it scales easily from the univariate (Beta) case to the multivariate (Dirichlet) case with the number of parameters only increasing by one per dimension. We found that the Bayesian approach can add accuracy and flexibility when working with this distribution family.

9 References

Adler D, Murdoch D, 2012, rgl: 3D visualization device system (OpenGL).
[http://CRAN.R-project.org/package=rgl]

Frigyik BA, Kapila A, Gupta MR, 2010, Introduction to the Dirichlet Distribution and Related Processes, UWEE Technical Report.
[http://www.ee.washington.edu/research/guptalab/publications/UWEETR-2010-0006.pdf]

Gelman A, Carlin JB, Stern HS, Rubin DB, 2004, Bayesian Data Analysis, 2nd Ed., Chapman & Hall.

Huang J, 2008, Maximum Likelihood Estimation of Dirichlet Distribution Parameters.
[http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.93.2881]

Jeffreys H, 1961, Theory of Probability, Oxford University Press.

Kotz S, Nadarajah S, 2000, Extreme Value Distributions, Imperial College Press.

Matsui T, Motoki M, Kamatani N, Kijima S, 2010, Polynomial time approximate or perfect samplers for discretized Dirichlet distribution, Japan Journal of Industrial and Applied Mathematics, 27, 91-123.

Minka TP, 2012, Estimating a Dirichlet distribution, Microsoft Research Paper.
[http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/minka-dirichlet.pdf]

Narayanan A, 1992, A note on parameter estimation in the multivariate beta distribution, Computational Mathematics and Applications, 24 (10), 11-17.

R Core Team, 2012, R: A language and environment for statistical computing, R Foundation for Statistical Computing. [http://www.R-project.org/]

Robert CP, Casella G, 2004, Monte Carlo Statistical Methods, 2nd Ed., Springer.

Ronning G, 1989, Maximum-likelihood estimation of Dirichlet distributions, Journal of Statistical Computation and Simulation, 32, 215–221.
[http://www.tandfonline.com/doi/abs/10.1080/00949658908811178]

Schafer JL, Olsen MK, 1998, Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst's Perspective, Multivariate Behavioral Research, 33 (4), 545-571.
[http://www.tandfonline.com/doi/abs/10.1207/s15327906mbr3304_5]

Von Hippel PT, 2012, Should a Normal Imputation Model be Modified to Impute Skewed Variables?, Sociological Methods & Research.
[http://smr.sagepub.com/content/early/2012/11/01/0049124112464866.abstract]

Wicker N, Muller J, Kalathur RKR, Poch O, 2008, A maximum likelihood approximation method for Dirichlet's parameter estimation, Computational Statistics & Data Analysis, 52 (3), 1315-1322.

Yang R, Berger JO, 1998, A Catalog of Noninformative Priors.
[http://www.stats.org.uk/priors/noninformative/YangBerger1998.pdf]

Zellner A, 1971, An Introduction to Bayesian Inference in Econometrics, John Wiley & Sons.

10 Appendix - Programming Code

This is the technical report version of this article; the published version will not list the code, but will instead have a link to this version.

10.1 dirichlet.r

The core functions for working with the Dirichlet distribution.

```
# Dirichlet distribution, by Sean van der Merwe

rdirichlet <- function(N=1,K=c(1,1)) {
# Simulations from the Dirichlet Distribution, according to the method of Wikipedia
lk <- length(K)
sim <- matrix(0,N,lk)
gams <- matrix(0,N,lk)
for (i in 1:lk) {
gams[,i] <- matrix(rgamma(N,K[i]),N,1)
}
gamtotal <- matrix(rowSums(gams),N,lk)
sim <- gams/gamtotal
return(sim)
}

psiinv <- function(y) {
x <- matrix(0,8,1)
if (y >= -2.22) {
    x[1] <- exp(y)+0.5
} else {
    x[1] <- -1/(y-digamma(1))
}
for (i in 2:8) {
    x[i] <- x[i-1] - ((digamma(x[i-1])-y)/trigamma(x[i-1]))
}
xf <- x[8]
return(xf)
}

dirichmom <- function(X) {
# Method of Moments estimates of the Dirichlet Distribution
temp <- dim(X); n <- temp[1]; m <- temp[2]
# X <- cbind(X,matrix(1-apply(X,1,sum)))
lpb <- apply(log(X),2,mean)
mom <- apply(X,2,mean)*(mean(X[,1])-mean(X[,1]^2))/(mean(X[,1]^2) - ((mean(X[,1]))^2))
return(matrix(mom))
```

```
}

dirichML <- function(X,mom=dirichmom(X),st=180) {
# Maximum Likelihood estimates of the Dirichlet Distribution
temp <- dim(X); n <- temp[1]; m <- temp[2]
# X <- cbind(X,matrix(1-apply(X,1,sum)))
lpb <- apply(log(X),2,mean)
a <- matrix(1,st,m)
a[1,] <- t(mom)
for (i in 2:st) {
    ai <- sum(a[i-1,])
    for (j in 1:m) {
        a[i,j] <- psiinv(digamma(ai) + lpb[j])
    }
}
ml <- matrix(a[st,],m)
return(ml)
}

deltadirichlet <- function(X,k,prior) {
k0 <- sum(k)
if (prior=='jeffreys') {
a <- 1 - trigamma(k0)*sum(trigamma(k)^(-1))
b <- psigamma(k0,2)*sum(trigamma(k)^(-1)) - trigamma(k0)*psigamma(k,2)/(trigamma(k)^2)
delta <- (nrow(X)*(digamma(k0)-digamma(k))) + colSums(log(X)) + 0.5*(psigamma(k,2)/trigamma(k)
- b/a)
} else {
delta <- (nrow(X)*(digamma(k0)-digamma(k))) + colSums(log(X)) + ((k-1)*trigamma(k)) -
(trigamma(k0)*(k0-length(k)))
}
return(delta)
}

lpostdirichlet <- function(X,k,prior='mdi') {
if (any(k<=0)) {
return(-Inf)
} else {
k0 <- sum(k)
if (prior=='jeffreys') {
        a <- nrow(X)*(lgamma(k0) - sum(lgamma(k)))
        b <- sum(log(trigamma(k))) + log(1 - trigamma(k0)*sum(trigamma(k)^(-1)))
        lpost <- a + (0.5*b) + sum((k-1)*colSums(log(X)))
} else {
        a <- (nrow(X) + 1)*(lgamma(k0) - sum(lgamma(k)))
        b <- (digamma(k)-digamma(k0)) + colSums(log(X))
        lpost <- a + sum((k-1)*b)
}
return(lpost)
}
}

dirichletpostmode <- function(X,prior='mdi',maxiter=4000,delt=0.1,deltadj=0.5) {
n <- nrow(X)
m1 <- ncol(X)
i <- 0
x1 <- rep(1,m1)
x2 <- dirichmom(X)
while ((i < maxiter) && (sum(abs(x2-x1))>(m1*10^-15))) {
    x1 <- x2
    i <- i + 1
    x2 <- x1 + delt*deltadirichlet(X,x1,prior)
    while (lpostdirichlet(X,x2,prior) < lpostdirichlet(X,x1,prior)) {
        delt <- delt * deltadj
        x2 <- x1 + delt*deltadirichlet(X,x1,prior)
    }
}
# cat(i,delt)
return(x2)
}

findsds <- function(X,topk,prior) {
topl <- lpostdirichlet(X,topk,prior)
m1 <- length(topk)
sds <- rep(1,m1)
for (i in 1:m1) {
        botk <- topk
```

```
        botk[i] <- topk[i]*0.5
        botl <- topl - lpostdirichlet(X,botk,prior) - 0.25
        midk <- topk
        midk[i] <- topk[i]*0.99
        midl <- -(topl - lpostdirichlet(X,midk,prior) - 0.25)
        newl <- botl
        while (abs(newl) > 0.001) {
                newk <- (botl*midk+midl*botk)/(botl+midl)
                newl <- topl - lpostdirichlet(X,newk,prior) - 0.25
                if (newl > 0) {
                        botk <- newk
                        botl <- newl
                } else {
                        midk <- newk
                        midl <- -newl
                }
        }
        sds[i] <- topk[i] - newk[i]
}
return(sds)
}

dirichletpostsim <- function(X,nsim=1000,prior='mdi',burnin=100) {
n <- nrow(X)
m1 <- ncol(X)
k1 <- dirichletpostmode(X,prior)
sds <- findsds(X,k1,prior)
accepted <- 0
tried <- 0
totsim <- nsim*2 + burnin
sims <- matrix(1,totsim,m1)
lp1 <- lpostdirichlet(X,k1,prior)
while (accepted < totsim) {
        k2 <- k1 + rnorm(m1)*sds
        while (any(k2<=0)) { k2 <- k1 + rnorm(m1)*sds*0.9 }
        lp2 <- lpostdirichlet(X,k2,prior)
        tried <- tried + 1
        if (lp2 - lp1 > log(runif(1))) {
                accepted <- accepted + 1
                k1 <- k2
                lp1 <- lp2
                sims[accepted,] <- k2
        }
}
# cat('\n Acceptance Rate: ',(accepted/tried),' \n')
sims <- sims[(-burnin:-1),]
return(sims[(1:nsim)*2,])
}

dirichletpostsimold <- function(X,nsim=1000,burnin=100,jumpsize=0.1) {
n <- nrow(X)
m1 <- ncol(X)
k1 <- log(dirichletpostmode(X))
accepted <- 0
tried <- 0
totsim <- nsim*2 + burnin
sims <- matrix(1,totsim,m1)
lp1 <- lpostdirichlet(X,exp(k1))
while (accepted < totsim) {
        k2 <- k1 + rnorm(m1)*jumpsize
        lp2 <- lpostdirichlet(X,exp(k2))
        tried <- tried + 1
        if (lp2 - lp1 > log(runif(1))) {
                accepted <- accepted + 1
                k1 <- k2
                lp1 <- lp2
                sims[accepted,] <- exp(k2)
        }
}
# cat('\n Acceptance Rate: ',(accepted/tried),' \n')
sims <- sims[(-burnin:-1),]
return(sims[(1:nsim)*2,])
}

dirichletpostsimfast <- function(X,prior='mdi',nsim=2000,burnin=100) {
m1 <- ncol(X)
```

```
k1 <- dirichletpostmode(X,prior)
sds <- findsds(X,k1,prior)
accepted <- 0
tried <- 0
totsim <- nsim + burnin
maxsim <- totsim*1000
sims <- matrix(1,totsim,m1)
lp1 <- lpostdirichlet(X,k1,prior)
while ((accepted < totsim) && (tried < maxsim)) {
        k2 <- k1 + rnorm(m1)*sds
        tried <- tried + 1
        while ((any(k2<=0)) && (tried < maxsim)) {
                k2 <- k1 + rnorm(m1)*sds*0.8
                tried <- tried + 1
        }
        if (tried < maxsim) {
                lp2 <- lpostdirichlet(X,k2,prior)
                if (lp2 - lp1 > log(runif(1))) {
                        accepted <- accepted + 1
                        k1 <- k2
                        lp1 <- lp2
                        sims[accepted,] <- k2
                }
        }
}
if (tried < maxsim) {
        sims <- sims[(-burnin:-1),]
        return(sims)
} else {
        return('single tear')
}
}
```

10.2 Simulation Study Code

```
getrmspe <- function(vec,acc = 5000) {
samsize <- vec[1]
truek <- vec[-1]
source('dirichlet.r')
spe <- matrix(0,acc,6)
for (i in 1:acc) {
        sam <- rdirichlet(samsize,truek)
        spe[i,1] <- mean(((dirichmom(sam)-truek)/truek)^2)
        spe[i,2] <- mean(((dirichML(sam)-truek)/truek)^2)
        spe[i,3] <- mean(((dirichletpostmode(sam,'mdi')-truek)/truek)^2)
        spe[i,4] <- mean(((dirichletpostmode(sam,'jeffreys')-truek)/truek)^2)
        spe[i,5] <- mean(((colMeans(dirichletpostsimfast(sam,'mdi'))-truek)/truek)^2)
        spe[i,6] <- mean(((colMeans(dirichletpostsimfast(sam,'jeffreys'))-truek)/truek)^2)
}
rmspe <- sqrt(colMeans(spe))
return(rmspe)
}

combinations <-
list(c(90,3,3,3),c(90,3,3,3,3,3),c(120,3,3,3,3,3,3,3,3),c(25,4,4,4),c(25,100,100,100),c(25,4,2
0,100),c(50,4,4,4),c(50,100,100,100),c(50,4,20,100),c(90,4,4,4),c(90,100,100,100),c(90,4,20,10
0),c(25,3,3,3),c(25,3,3,3,3,3),c(50,3,3,3),c(50,3,3,3,3,3))
ncombi <- length(combinations)
descrip <- rep(' ',ncombi)
for (i in 1:ncombi) {descrip[i] <- paste('n=',combinations[[i]][1],';
k=(',paste(combinations[[i]][-1],collapse=','),')',sep='')}
library(parallel)
cl <- makeCluster(4)
system.time(out <- parLapplyLB(cl,combinations,getrmspe,5000))
stopCluster(cl)
results <- t(matrix(unlist(out),6,ncombi))
estimators <- c('MOM','ML','ModeMDI','ModeJeffreys','MeanMDI','MeanJeffreys')
rownames(results) <- descrip
colnames(results) <- estimators
write.csv(results,'simstudy1p1.csv')
combined <- colMeans(results)
colnames(combined) <- estimators
print(combined)
```

```
write.csv(combined,'simstudy1p3.csv')
print(estimators[order(combined)])
write.csv(estimators[order(combined)],'simstudy1p4.csv')
getwinner <- function(measures) {return(estimators[which.min(measures)])}
winners <- apply(results,1,getwinner)
results <- cbind(results,winners)
colnames(results) <- c(estimators,'Best Estimator')
results
write.csv(results,'simstudy1p2.csv')
```

10.3 Iris Examples Code

```
library(rgl)
data(iris)
attach(iris)
species <- levels(Species)
source('dirichlet.r')
k2 <- matrix(1,3,5)
k3 <- matrix(1,3,5)
samples <- vector('list',3)
for (i in 1:3) {
        si <- as.matrix(iris[Species==species[i],1:4])
        si <- si/matrix((1+rowSums(si)),50,4)
        samples[[i]] <- cbind(si,matrix((1-rowSums(si)),50))
        k2[i,] <- dirichletpostmode(samples[[i]])
}
k2

sim11 <- rdirichlet(50,k2[1,])
sim12 <- rdirichlet(50,k2[2,])
sim13 <- rdirichlet(50,k2[3,])
sim31 <- sim11[,1:4]/(1-rowSums(sim11[,1:4]))
sim32 <- sim12[,1:4]/(1-rowSums(sim12[,1:4]))
sim33 <- sim13[,1:4]/(1-rowSums(sim13[,1:4]))
sim2 <- rbind(sim31,sim32,sim33)
plot3d(c(Sepal.Length,sim2[,1]),c(Sepal.Width,sim2[,2]),c(Petal.Length,sim2[,3]),type="s",size
=c(Petal.Width,sim2[,4]),col=rep(c('#000044','#440000','#004400','#0000CC','#CC0000','#00CC00'
),each=50),main="Iris data with simulated replicates",xlab="Sepal Length",ylab="Sepal
Width",zlab="Petal Length",sub="Size indicates Petal Width")

ver <- samples[[2]]
T <- 9999
K <- dirichletpostsim(ver,T)
colK <- cbind(K[,c(3,4)],rowSums(K[,c(1,2,5)]))
Ys <- matrix(0,T,2)
for (i in 1:T) {
        u1 <- c(rdirichlet(1,colK[i,]))
        y1 <- 10*u1[1:2]/u1[3]
        Ys[i,] <- y1
}
colnames(Ys) <- c('Petal Length','Petal Width')
colMeans(Ys)
library(rgl)
demo(hist3d)
rgl.open()
rgl.bg(color="white")
hist3d(Ys[,1],Ys[,2],col="#00aa00",nclass=15,scale=30)
box3d()
axis3d('x')
axis3d('z')
title3d('Predictive Posterior Conditional Distribution','','Petal Length','','Petal Width')
```

10.4 Rainfall Imputation Code

```
setwd('C:\\Work\\Work2013\\Research\\Dirichlet')
rain <- read.csv('rain.csv',row.names=1)
source('dirichlet.r')
full <- rain[1:37,]
holey <- rain[38:95,]
fulld1 <- full/(1+rowSums(full))
fulld1 <- cbind(fulld1,(1-rowSums(fulld1)))
finalests <- matrix(0,11,6)
```

```
finalests[1,] <- dirichletpostmode(fulld1)
for (dataset in 1:10) {
K <- dirichletpostsim(fulld1,1)
imputed <- holey
for (r in 1:40) {
for (i in 1:58) {
miss <- which(is.na(holey[i,]))
notmiss <- 1:5
notmiss <- notmiss[-miss]
colK <- cbind(K[miss],sum(K[c(notmiss,6)]))
u1 <- c(rdirichlet(1,colK))
filling <- u1[1]/u1[2]*(1+sum(imputed[i,notmiss]))
imputed[i,miss] <- filling
}
alldata <- rbind(full,imputed)
alldatad1 <- alldata/(1+rowSums(alldata))
alldatad1 <- cbind(alldatad1,(1-rowSums(alldatad1)))
K <- dirichletpostsim(alldatad1,1)
}
finalests[(1+dataset),] <- dirichletpostmode(alldatad1)
write.csv(alldata,paste('rainimp',dataset,'.csv',sep=''))
}
write.csv(finalests,'rainests.csv')
```