Dirichlet - GLM model for analyzing fractional data

Lee, Woo Jung
ORF 376: Junior Independent Work
Department of Operations Research and Financial Engineering,
Princeton University
wool@princeton.edu

Supervised by
Professor Barbara Engelhardt
Department of Computer Science,
Princeton University
bee@cs.princeton.edu

May 5, 2015

Contents

Abstract

Fractional data, also known as compositional data, is multivariate data whose components sum up to 1. It can represent useful quantities such as proportions and probabilities. Analysis of fractional data is hindered by the heteroscedasticity inherent among the components. This paper aims to develop a predictive model for such data based on the theories of generalized linear model, using Dirichlet distribution as the assumed distribution of response variables. Beta distribution, which is a special case of Dirichlet distribution, is examined first using similar approach; then, it is generalized to our Dirichlet regression model. Parameter fitting is done by maximum likelihood estimation with respect to the linear coefficients. This model can handle heteroscedasticity and thus performs better compared to widely used logit-transformation.

Python module DirichletRegression implements this model on a framework based on generalized_linear_model in statsmodels module. Implementation and computation are discussed, along with required parameters for the model. This module is very flexible, as it allows users to decide computation methods used in the model.

Keywords: Dirichlet Distribution, Generalized Linear Model, Beta Distribution, Fractional Data

1 Introduction

Compositional data arise in many fields from geology to economics; it can represent proportion of certain things, probability distribution of certian variable, or even be used in complex machine learning fields such as topic models. Despite the great abundance of compositional data, the analysis of such data is difficult due to the unique covariance structure among the component imposed by the unit sum constraint. Note that the sum constraint does not have to be unit sum; we can simply transform the data set so that it fits within $[0, 1]$. Aitchinson [1], in his book The Statistical Analysis of Compositional Data, explores many different methods of dealing with compositional data; however, the heteroscedasticity is not easy to ignore. With emergence

of fast computers and advanced algorithms, we can develop better models that can handle heteroscedasticity. When Dirichlet distribution is assumed as distribution of the response variables in regression model, we can use this Dirichlet-GLM model to do regression analysis on the data. Although generalized linear model is used widely for simpler models with one or two parameters, it can still be extended to case of Dirichlet distribution as Dirichlet distribution belongs in the exponential distribution family; we will see that the choise of this distribution family allows us to perform maximum likelihood estimation and build regression mdoel.

The model trains itself on observations of covariates and response variables; then, it accepts a vector as input and make a prediction value, which belongs to Dirichlet distribution. During the fitting process, picking starting values and optimization methods are crucial as they determine the speed and accuracy of the model.

Although generalized linear model is used very frequently in many fields, it does not get much attention when the response variable has more than 1 variable. Ferrari and Cribari-Neto[**?**] develop model called beta regression in their paper. BecauseDirichlet distribution is a generalization of beta distribution (when the dimension of Dirichlet distribution is 2, it is the same as beta distribution), the beta regression model can be extended to arbitrary dimension D with Dirichlet distribution instaed of beta distribution.

One practical problem is the fact that Dirichlet distribution has support of $(0, 1)^D$ instead of $[0, 1]^D$. Therefore, if there are 0s or 1s, we need to make appropriate changes to the values to assure that our model performs fine. Bad choice of transformation could strongly affect the covariance structure of different components, leading to undesirable results.

In this paper, we review papers that study beta regression and Dirichlet regression, and aim to develop a Python package `DirichletDistribution` that is flexible and fast. One focus would be on allowing various methods to be available to the users so they can pick the best method depending on

their interest and data set.

First, we will examine the properties of beta distribution and beta regression.

2 Dirichlet-GLM model

Dirichlet distribution is multivariate generalization of beta distribution; beta distribution models values in $(0, 1)$, and Dirichlet distribution models multivariate extension of this distribution to multivariate case, with the unit sum constraint on the components. Therefore, examining some properties of beta distribution, we can get a better understanding of our Dirichlet-GLM model.

2.1 Beta distribution

We will examine beta distribution first, and make a generalization to Dirichlet distribution. Probability density function of beta distribution is given by:

$$f(x|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1},\ \alpha, \beta > 0 \tag{1}$$

where $\Gamma(\cdot)$ denotes gamma function, $x \in (0, 1)$, and $\alpha > 0$, $\beta > 0$. Its mean and variance are:

$$\mathbb{E}(x) = \frac{\alpha}{\alpha + \beta} \qquad \text{Var}(x) \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \tag{2}$$

Beta distribution is good at modeling data that lies in the range $(0, 1)$ because of its flexibility. The two shape parameters α and β determine the shape of the distribution, and it enables us to model even heavily skewed data. In his paper, Ferrari develops beta regression model, which has generalized linear model style structure. Link function is used to relate the linear predictor, $\boldsymbol{X\beta}$, to the parameters of beta distribution, fitted using maximum

likelihood estimation. Although Ferrari focuses more on re-parameterized beta distribution rather than the classic parametrization (as basis for generalization to Dirichlet distribution), similar method of parameter fitting can be adopted to our case.

Beta distribution will also be useful in deriving starting values for our optimization principle. One of the provided methods for starting value selection in DirichletRegression module is done under the following theorem:

Theorem 2.1. *Marginal distribution of Dirichlet distribution is beta distribution. That is, if* $(X_1, X_2, X_3) \sim Dirichlet(\alpha_1, \alpha_2, \alpha_3)$ *Then,* $X_1 \sim Beta(\alpha_1, \alpha_2 + \alpha_3)$.

Proof. The probability density function of (X_1, X_2, X_3) is (as shown in equation (3) below):

$$f_{X_1,X_2,X_3}(x_1, x_2, x_3) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1 - 1} x_2^{\alpha_2 - 1} (1 - x_1 - x_2)^{\alpha_3 - 1}$$

The probability density function of the marginal distribution of X_1 is:

$$\begin{aligned}
f_{X_1}(x_1) &= \int_{-\infty}^{\infty} f_{X_1,X_2,X_3}(x_1, x_2, x_3) dx_2 \\
&= \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1 - 1} \int_0^{1 - x_1} x_2^{\alpha_2 - 1} (1 - x_1 - x_2)^{\alpha_3 - 1} dx_2 \\
&= \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1 - 1} (1 - x_1)^{\alpha_2 + \alpha_3 - 1} \int_0^1 y^{\alpha_2 - 1} (1 - y)^{\alpha_3 - 1} dy \\
&= \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} x_1^{\alpha_1 - 1} (1 - x_1)^{\alpha_2 + \alpha_3 - 1} \frac{\Gamma(\alpha_2)\Gamma(\alpha_3)}{\Gamma(\alpha_2 + \alpha_3)} \\
&= \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2 + \alpha_3)} x_1^{\alpha_1 - 1} (1 - x_1)^{\alpha_2 + \alpha_3 - 1}
\end{aligned}$$

Therefore, $X_1 \sim Beta(\alpha_1, \alpha_2 + \alpha_3)$. □

Using this theorem repeatedly for Dirichlet distribution of any dimension D, we see that if $\boldsymbol{Y} \sim Dirichlet(\boldsymbol{\alpha})$, then $y_i \sim Beta(\alpha_i, \sum_{i \neq i} \alpha_i)$. Using this property, we will estimate starting values for each α separately in our algorithm. More details about this implementation will be discussed later.

2.2 Dirichlet distribution

Let $\boldsymbol{Y} \sim Dirichlet(\boldsymbol{\alpha})$, whose dimension is D. Its probability density function is

$$f(y_1, ..., y_{D-1}; \alpha_1, ..., \alpha_D) = \frac{\Gamma(\sum_{i=1}^{D} \alpha_i)}{\sum_{i=1}^{D} \Gamma(\alpha_i)} \prod_{i=1}^{D} y_i^{\alpha_i - 1}, \ \boldsymbol{\alpha} > 0 \tag{3}$$

Note that Dth component of $\boldsymbol{y}$ does not need to be specified because $y_D = 1 - (\sum_{i=1}^{D-1} y_i)$. Parameter is $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_D)$, and moments are:

$$\mathbb{E}(y_i) = \frac{\alpha_i}{\alpha_T} \qquad \text{Var}(y_i) = \frac{\alpha_i(\alpha_T - \alpha_i)}{\alpha_T^2(\alpha_T + 1)} \qquad \text{Cov}(y_i, y_j) = -\frac{\alpha_i \alpha_j}{\alpha_T^2(1 + \alpha_T)} \tag{4}$$

where $\alpha_T = \sum_{i=1}^{D} \alpha_i$.

Because of its unique simplex support, Dirichlet distribution is used frequently in machine learning, often times as a prior distribution in Bayesian setting as it is conjugate prior of multinomial distribution. Therefore, if we choose prior of multinomial parameters to be Dirichlet distribution, the posterior distribution is also Dirichlet-distributed. This allows one to determine how much our belief about parameters changed after the observations were made.

Below are three plots of Dirichlet distribution, for parameters $\alpha = (0.9, 0.9, 0.9), (5, 5, 5), (2, 10, 20)$, respectively. Red region indicates higher concentration of points; we see that when $\alpha < 1$, as in first plot, data points are more centered around the corners, whereas when $\alpha > 1$, data points are more concentrated around the center. Last plot shows confirms this: we see that higher the value of α_i is, the contour peak occurs closer to corner of that index. Allso note that we can apply Dirichlet distribution to any compositional data in any range. For instance, given observation $\boldsymbol{y}_i = (y_1, \cdots, y_D)$ with $\sum_i y_i = C$, which lies on $[\min(\boldsymbol{y}_i), \max(\boldsymbol{y}_i)]^D$. We can transform the data from $\boldsymbol{y}_i$ to $\boldsymbol{y}x_i* = \frac{\boldsymbol{y}_i - \min(\boldsymbol{y}_i)}{\max(\boldsymbol{y}_i) - \min(\boldsymbol{y}_i)}$ so that the new variable $\boldsymbol{y}x_i*$ lies in $[0, 1]^D$. We will see further on that additional transformation is required for our model, which will be discussed later.

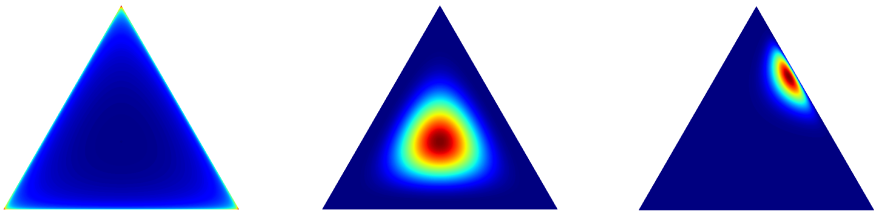

Figure 1: Ternary plots showing contours of Dirichlet distribution with parameters $\alpha = (0.9, 0.9, 0.9), (5, 5, 5), (2, 10, 20)$, respectively. Bottom axis is component 3, left axis is component 2, and right axis is component 1.

2.3 Generalized linear model and extension to Dirichlet distribution

Generalized linear model is widely used because of its flexibility in terms of response variable. It can take any covariate as input and predict values for response variables. It is generalization of ordinary least squares, which takes form of $\boldsymbol{Y} = \boldsymbol{X\beta}$, to a regression model that can handle any type of response variable that assumes a distribution in exponential distribution family by using link function. Given the covariate matrix $\boldsymbol{X}$, which is $n \times p$ matrix whose row $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ represents ith observation. The main features of this algorithm is:

1. Response variable that belongs to exponential distribution family: $\boldsymbol{Y}$
2. Linear predictor, $\boldsymbol{\eta} = \boldsymbol{X\beta}$
3. Link function, which is monotonic and differentiable: $g(\cdot)$
4. Mean expected response, $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{y}] = g^{-1}(\boldsymbol{\eta})$
5. Variance as a function of mean, $V(\boldsymbol{\mu})$

This model is extensively studied in *Generalized Linear Models* by McCullagh and Nelder[7]. It uses characteristics of exponential distribution family to; one of them is the concavity of the ln likelihood function. The probability

density function of multivariate exponential distribution family is:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = h(\boldsymbol{x})\exp(b(\theta)\cdot T(\boldsymbol{x}) - A(\theta)) \tag{5}$$

where $\mathbf{x}$ is covariate, θ is natural parameter. If $b(\theta) = \theta$, we say that the exponential distribution is canonical. The following theorem allows us to fit the model using maximum likelihood estimation.

Theorem 2.2. *Maximum likelihood function of exponential distribution is concave.*

Proof. Theorem 1.6.3 from [2] proves that $A(\cdot)$ is convex. The log-likelihood is

$$\mathcal{L}(\boldsymbol{\theta}) = \log\left(h(\boldsymbol{x})\right) + \sum_{i=1}^{D} b(\boldsymbol{\theta})\cdot T(\boldsymbol{x}) - A(\boldsymbol{\theta})$$

Because A is convex, $-A$ is concave. Therefore, as $\mathcal{L}$ is sum of linear function of θ and A is concave, $\mathcal{L}$ is concave. □

Dirichlet distribution is in exponential distribution family, with

- $\boldsymbol{\theta} = \boldsymbol{\alpha}$
- $h(\boldsymbol{x}) = 1$
- $b(\boldsymbol{\theta}) = \boldsymbol{\alpha} - 1$
- $T(\boldsymbol{x}) = \ln(\boldsymbol{x})$
- $A(\boldsymbol{\theta}) = \sum_{i=1}^{D} \ln\Gamma(\alpha_i) - \ln\Gamma(\sum_{i=1}^{D}\alpha_i)$

Therefore, we can use maximum likelihood estimation to fit our model. In our Dirichlet-GLM model, we assume that the response variable $\boldsymbol{Y}$ assumes Dirichlet distribution, $Dirichlet(\boldsymbol{\alpha})$. Given covariates, we create matrix $\boldsymbol{X}$ whose row $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \cdots, x_{iD})$ represents a single observation. Therefore, $\boldsymbol{X}$ is $n \times p$, where p denotes number of covariates. $\boldsymbol{Y}$ will be $n \times D$, whose row is observed response variable of dimension D. Thus, $\boldsymbol{\beta}$ will be $p \times D$, and this will be the parameters that we wish to fit. We will use log function as link function, so that $\boldsymbol{\alpha} > 0$. In other words, $\boldsymbol{\alpha} = g^{-1}(\boldsymbol{\eta}) = \exp(\boldsymbol{\eta})$ $\boldsymbol{\alpha}$, which will be used while fitting $\boldsymbol{\beta}$, will be $n \times D$.

2.4 Preprocessing data

Before we can move on to parameter estimation, we need to make transformations to the values that take form of 0 and 1. The support of Dirichlet distribution is $(0,1)^D$, and therefore in the case in which there are values outside this range, we should perform one of the following suggested methods. Furthermore, if the value is below the detection thereshold level of the computer, it will be considered as 0 and cause problems. Therefore, we need to make appropriate corrections to these extreme values.

Aitchison in his book [1] suggests method in which all the zeros are replaced with certain small value δ. However, this is not the most optimal method for our model because it distorts the covariance structure inherent in the response variables. Martin-Fernandez proposed multiplicative method in his paper [5]: for each row $\boldsymbol{y_i}$ with c zeros,

$$y_{ij} = \begin{cases} \delta & \text{if } y_{ij} \leq 0 \\ (1 - c\delta)y_{ij} & : \text{if } y_{ij} > 0 \end{cases}$$

It is also shown that using δ of around 65% of the detection level yields the best result. However, the final result is too sensitive towards the value of δ; also, this method replaces every zero element with same value.

Method suggested by Smithson [8] replaces every zero element with the following transformation:

$$y_{ij} = \frac{y_{ij}(n-1) + 1/D}{n}$$

where n is the number of observations. (This method is used for beta regression in Smithson's paper, so his version has term $1/2$ on numerator instead of $1/D$. This is generalized case of the formula in D-dimension.) This transformation shrinks the support from $[0,1]$ to $(\frac{n}{D}, 1 - \frac{D-1}{nD})$, which converges to $(0,1)$ as $n \to \infty$ Although this method has a drawback that all the replaced values are the same, it performs well under large data; this transformation shrinks the support from $[0,1]^D$ to $(\frac{1}{nD}, 1 - \frac{D-1}{nD})^D$, which converges to $(0,1)^D$ as $n \to \infty$. This is the transformation method that we use in our DirichletRegression module.

2.5 Parameter estimation

The log-likelihood function of Dirichlet distribution is

$$\mathcal{L}(\boldsymbol{\alpha}|\boldsymbol{X},\boldsymbol{Y}) = \sum_{i=1}^{n}\Big[\log\Gamma(\alpha_{iT}) - \sum_{j=1}^{D}\log\Gamma(\alpha_{ij}) + \sum_{j=1}^{D}(\alpha_{ij}-1)\log y_{ij}\Big]$$

where $\alpha_{iT} = \sum_{j=1}^{D}\alpha_{ij}$, and log is natural log. In our model, where we aim to estimate $\boldsymbol{\alpha}$ using $\boldsymbol{\beta}$, we have:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}|\boldsymbol{X},\boldsymbol{Y}) &= \sum_{i=1}^{n}\mathcal{L}_i(\boldsymbol{\beta}|\boldsymbol{X},\boldsymbol{y_i}) \\ &= \sum_{i=1}^{n}\Big[\log\Gamma\Big(\sum_{j=1}^{D}\exp(\eta_{ij})\Big) \\ &\qquad - \sum_{j=1}^{D}\Big(\log\Gamma(\exp(\eta_{ij})) - (\exp(\eta_{ij})-1)\log y_{ij}\Big)\Big]\end{aligned}$$

where $\exp(\eta_{ij}) = \alpha_{ij}$.

Using theorem 2.2, we see that $\hat{\boldsymbol{\beta}}$ is maximum likelihood estimate of the above function if and only if it maximizes it. Therefore, we need to derive expressions for first and second order derivative of the log likelihood function. Gradient vector is

$$\mathcal{G}(\boldsymbol{\beta}|\boldsymbol{X},\boldsymbol{Y}) = \Big[\frac{\partial\mathcal{L}}{\partial\beta_{11}}, \frac{\partial\mathcal{L}}{\partial\beta_{12}}, \cdots, \frac{\partial\mathcal{L}}{\partial\beta_{1D}}, \frac{\partial\mathcal{L}}{\partial\beta_{21}}, \frac{\partial\mathcal{L}}{\partial\beta_{22}}, \cdots, \frac{\partial\mathcal{L}}{\partial\beta_{pD}}\Big] \tag{6}$$

where β_{ij} is row i column j of $\boldsymbol{\beta}$. Calculating each derivative requires chain rule:

$$\frac{\partial\mathcal{L}_i}{\partial\alpha_{ij}} = \psi(\alpha_{iT}) - \psi(\alpha_{ij}) + \log y_{ij},$$
$$\frac{d\alpha_{ij}}{d\eta_{ij}} = \exp(\eta_{ij}) = \alpha_{ij}, \qquad \frac{\partial\eta_{ij}}{\partial\beta_{kj}} = x_{ik}$$

$$\therefore \quad \frac{\partial \mathcal{L}}{\partial \beta_{kj}} = \frac{\partial}{\partial \beta_{kj}} \sum_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\beta}|\boldsymbol{y_i}) \tag{7}$$

$$= \sum_{i=1}^{n} \frac{\partial \mathcal{L}_i}{\partial \beta_{kj}} \tag{8}$$

$$= \sum_{i=1}^{n} \Big[\frac{\partial \mathcal{L}_i}{\partial \alpha_{ij}} \frac{d\alpha_{ij}}{d\eta_{ij}} \frac{\partial \eta_{ij}}{\partial \beta_{kj}} \Big] \tag{9}$$

$$= \sum_{i=1}^{n} (\psi(\alpha_{iT}) - \psi(\alpha_{ij}) + \log y_{ij}) \alpha_{ij} x_{ik} \tag{10}$$

where ψ is digamma function, $\psi(x) = d \log \Gamma(x)/dx$. Hessian matrix is as following:

$$\mathcal{H}(\boldsymbol{\beta}) = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial \beta_{11}^2} & \frac{\partial^2 \mathcal{L}}{\partial \beta_{11} \partial \beta_{12}} & \cdots & \frac{\partial^2 \mathcal{L}}{\partial \beta_{11} \partial \beta_{p(D-1)}} & \frac{\partial^2 \mathcal{L}}{\partial \beta_{11} \partial \beta_{pD}} \\ 0 & \frac{\partial^2 \mathcal{L}}{\partial \beta_{12}^2} & & & \frac{\partial^2 \mathcal{L}}{\partial \beta_{12} \partial \beta_{pD}} \\ \vdots & & \ddots & & \vdots \\ 0 & & & \frac{\partial^2 \mathcal{L}}{\partial \beta_{p(D-1)}^2} & \frac{\partial^2 \mathcal{L}}{\partial \beta_{p(D-1)} \partial \beta_{pD}} \\ 0 & 0 & \cdots & 0 & \frac{\partial^2 \mathcal{L}}{\partial \beta_{pD}^2} \end{bmatrix}$$

where the rows and columns are the pD parameters and are indexed similarly to gradient vector (6). The second order derivative of the log-likelihood

function is:

$$\frac{\partial^2 \mathcal{L}}{\partial \beta_{kj} \partial \beta_{uv}} = \frac{\partial}{\partial \beta_{uv}} \Big(\frac{\partial \mathcal{L}}{\partial \beta_{kj}} \Big) \tag{11}$$

$$= \frac{\partial}{\partial \beta_{uv}} \Big(\sum_{i=1}^{n} \frac{\partial \mathcal{L}_i}{\partial \beta_{kj}} \Big) \tag{12}$$

$$= \sum_{i=1}^{n} \frac{\partial}{\partial \alpha_{iv}} \Big(\frac{\partial \mathcal{L}_i}{\partial \alpha_{ij}} \frac{d\alpha_{ij}}{d\eta_{ij}} \frac{\partial \eta_{ij}}{\partial \beta_{kj}} \Big) \frac{d\alpha_{iv}}{d\eta_{iv}} \frac{\partial \eta_{iv}}{\partial \beta_{uv}} \tag{13}$$

$$= \sum_{i=1}^{n} \frac{\partial}{\partial \alpha_{iv}} \Big(\frac{\partial \mathcal{L}_i}{\partial \alpha_{ij}} \frac{d\alpha_{ij}}{d\eta_{ij}} \Big) x_{ik} \alpha_{iv} x_{iu} \tag{14}$$

$$= \sum_{i=1}^{n} \Big(\frac{\partial^2 \mathcal{L}_i}{\partial \alpha_{ij} \partial \alpha_{iv}} \frac{d\alpha_{ij}}{d\eta_{ij}} + \frac{\partial \mathcal{L}_i}{\partial \alpha_{ij}} \frac{\partial}{\partial \alpha_{iv}} \frac{d\alpha_{ij}}{d\eta_{ij}} \Big) x_{ik} \alpha_{iv} x_{iu} \tag{15}$$

$$= \sum_{i=1}^{n} \mathscr{F}(i, j, v, \boldsymbol{\alpha}, \boldsymbol{\eta}) \alpha_{iv} x_{ik} x_{iu} \tag{16}$$

If $j = v$:

$$\mathscr{F}(i, j, v, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \alpha_{ij}(\psi'(\alpha_{iT}) - \psi'(\alpha_{ij})) + \psi(\alpha_{iT}) - \psi(\alpha_{ij}) + \log y_{ij}$$

If $j \neq v$:

$$\mathscr{F}(i, j, v, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \psi'(\alpha_{iT}) \alpha_{ij}$$

where ψ' is trigamma function, $\psi'(x) = d^2 \log \Gamma(x)/dx^2$. Therefore:

$$\frac{\partial^2 \mathcal{L}}{\partial \beta_{kj} \partial \beta_{uv}} = \begin{cases} \sum_{i=1}^{n} [\alpha_{ij}(\psi'(\alpha_{iT}) - \psi'(\alpha_{ij})) \\ \qquad + \psi(\alpha_{iT}) - \psi(\alpha_{ij}) + \log y_{ij})] x_{iu} \alpha_{iv} x_{ik} & \text{if } j = v \\ \sum_{i=1}^{n} \psi'(\alpha_{iT}) \alpha_{ij} \alpha_{iv} x_{ik} x_{iu} & \text{if } j \neq v \end{cases} \tag{17}$$

Fisher's information matrix can be written as $\mathcal{I}(\hat{\beta}) = \mathbb{E}(H(\hat{\beta}))$. Then, we have

$$\hat{\beta} \sim \mathcal{N}(\beta, \mathcal{I}^{-1}(\hat{\beta}))$$

Once we obtain maximum likelihood estimator using appropriate optimization algorithm discussed in section 3, we can get our final estimated parameters $\boldsymbol{\beta}$, which we can use to make prediction given a vector of size p. The most steps of the model are starting value computation and optimization choice, as it determines how fast, if at all, the optimization will converge.

3 DirichletRegression module implementation

DirichletRegression is a module based on Dirichlet-GLM model discussed in this paper. The code is based on statsmodels' generalized_linear_model module. It takes $\boldsymbol{X}$, observation matrix, $\boldsymbol{Y}$, response matrix, and offset/exposure as input. Offset/exposure is for adding constant intercept parameters to our model. Exposure can only be used with log link, which we use in our model extensively. For more information, see [7].

DirichletRegression module is largely divided up to 3 parts: preparing the data, making a good starting value input for optimization algorithms, and fitting parameters. Each step is explained and algorithms used are discussed. This module was developed on Python 3.4.

3.1 Preprocessing

As shown above, there exist many studies on dealing with extreme values in Dirichlet distributions. Method suggested by Smithson [8] is implemented: `_preprocess(self, endog, exog)` examines endogenous variable matrix ($\boldsymbol{Y}$) and replace extreme values that lie above or below the predetermined detection threshold level; that is,

$$y_{ij} = y_{ij} = \frac{y_{ij}(n-1) + 1/D}{n} \quad \text{if } y_{ij} < \texttt{FLOAT\_EPS} \text{ or } y_{ij} > 1 - \texttt{FLOAT\_EPS}$$

where c is the number of zeros in row $\boldsymbol{y}_i$, and `FLOAT_EPS` = `numpy.finfo(float).eps` in Python environment. This preprocessing takes care of what is called "rounded" zeros, which are caused by measurement and data collecting process. These are recognized as zero by computers, but doing so may affect the inherent structure of the model; therefore, it is a good idea to perform these transformations. [3]

3.2 Starting value estimation

As this model is in high dimension in accordance with n, p, and D, it is important to derive a good starting value so that the optimization algo-

rithms will converge. The module provides three different methods of computing starting values using the method `_starting_values(self, endog, exog, svmethod)` `svmethod` must be one of `meanprecision`, `fixedpoint`, or `beta`. The first two methods combine maximum likelihood estimator from `dirichlet` package, developed by Eric Suh ,[10] with sampling process coined by Hijazi.[4] Here is pseudo-algorithm of the sampling process, mixed with Dirichlet maximum likelihood estimation. `dirichlet` package returns fitted constant Dirichlet parameters, using either `meanprecision` or `fixedpoint` method. Sampling process by Hijazi allows us to estimate starting points using only constant Dirichlet model and ordinary least squares.

1. Choose k samples of size m, with replacement. $(m \leq n)$
2. For each sample, fit a Dirichlet model with constant parameter using `mle` method in `dirichlet`. Also, compute the mean of corresponding covariates.
3. This yields a matrix, $\boldsymbol{V}$, which has dimension of $k \times (D + p)$. The first D columns represent maximum likelihood estimates of k samples, and the last p columns represents means of the covariates in each sample.
4. Fit D least squares models using $V_i = g^{-1}(V_{D:D+p}), i = 1, \cdots, D$ where g is our link, log function.
5. Use the yielded coefficients as starting values.

Hijazi claims that simulation returned the highest probability of convergence when $m \approx \lceil n/3 \rceil$ with $k \approx 20$. These are the values that we use in `DirichletRegression`

Another method provided is `beta`. This method is used by Maier in his work, [6] and it is done by assuming that each dimension is beta-distributed, as we showed in theorem 2.1 earlier. For each dimension $j = 1, \cdots, D$, we compute the likelihood and gradient of beta distribution with jth component of observations, and use BFGS (BroydenFletcherGoldfarb-Shanno) algorithm to find the maximum likelihood estimates. The starting values are simply set as $(\alpha = 0.5, \beta = 0.5)$ for each beta distribution, with tolerance level of `1e-05`. This method performs much less calcuation then

dirichlet sampling method. Although it may perform poorly in terms of accuracy, it performs fast, and can be useful for small dataset.

3.3 Fitting

As the gradient and hessian values are not in closed form, we must use non-linear optimization method to fit the parameters. Two methods are usable with this module, along with combination of both. Using `_optimize(self, endog, exog, sv, optmethod)`, which extensively sues `minimize` method from `statsmodels` module. [9] Three possible specification of optimization methods is `BFGS`, `NCG`, or `both`. If one of the two methods is picked, only that method will be used. If `both`, the objective function is first optimized using BFGS with tolerance level of `1e-7`. Then, the final estimates are used as starting values for NCG (Newton-Conjugate Gradient) method. Higher tolerance level of `1e-13` is used for better convergence. `both` is the default method used.

Combgining the two optimization method yield much accurate model; using only BFGS method may not be ideal, as standard errors extracted from Hessian matrix are not too reliable, although BFGS algorithm is shown to be fast and robust. Running NCG algorithm in conjunction with BFGS allows better convergence and better estimate of standard errors: with NCG method, the standard errors are:

$$\begin{aligned} \mathrm{SE}(\hat{\theta}) &= \sqrt{\mathrm{diag}(\boldsymbol{\Sigma}(\hat{\theta}))} \\ &= \sqrt{\mathrm{diag}(-\mathcal{H}(\hat{\theta})^{-1})} \end{aligned}$$

3.4 Function parameters

To create a `DirichletRegression` object, the following command is used:

```
DR = DirichletRegression(endog, exog, svmethod='meanprecision',
optmethod='both', offset=None, exposure=None, missing='none', **kwargs)
```

with

endog	Response Variable Matrix	$n \times D$
exog	Covariate Matrix	$n \times p$
svmethod	Starting Value Computation Method.	One of: meanprecision, fixedpoint, or beta. Default is meanprecision.
optmethod	Optimization Method.	One of: BFGS, NCG, or both. Default is both.
offset, exposure	Extra Constant Term	See generalized_linear _models for more details
missing	What to do with missing values in matrix?	See generalized_linear _models for more details

3.5 Analysis

4 Discussions

This paper developed and explained the basic structure of Dirichlet regression. We also examined its implementation in Python language. The three main steps of this model, preprocess, starting value computation, and fitting, are elaborated. The main advantage of this model is that there is no necessity to transform the whole given data; we only need to make transformations on zeros or ones. Furthermore, it can inherently handle heteroscedasticity of the data.

Furthere studies on this topic can be done on finding better optimization or starting value computation method; one method propsed by Hijazi [3] based on EM algorithm could be used to derive starting values. Better form of model selection that measures how well each model handles skewedness of the data can be developed to analyze and improve the model.

References

[1] J. Aitchison. *The Statistical Analysis of Compositional Data*. The Blackburn Press, 2003.

[2] P. J. Bickel and K. A. Doksum. *Mathematical Statistics, Basic Ideas and Selected Topics*, volume 1. Pearson, second edition, may 2006.

[3] R. Hijazi, editor. *An EM-Algorithm Based Method to Deal with Rounded Zeros in Compositional Data under Dirichlet Models*, 2011. International Workshop on Compositional Data Analysis.

[4] R. H. Hijazi and R. W. Jernigan. Modelling compositional data using dirichlet regression models. *Journal of Applied Probability & Statistics*, 4(1):77–91.

[5] V. P.-G. J.A. Martin-Fernandez, C. Barcelo-Vidal. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. *Mathematical Geology*, 35(3), apr 2003.

[6] M. J. Maier. Dirichletreg: Dirichlet regression for compositional data in r. Research Report Series Report 125, Institute for Statistics and Mathematics, jan 2014.

[7] P. McCullagh and J. A. Nelder. *Generalized Linear Models, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Chapman and Hall/CRC, second edition, aug 1989.

[8] M. Smithson and J. Verkuilen. A better lemon squeezer? maximum-likelihood regression with beta-distributed dependent variables. *Psychological Methods*, 11(1):54–71, 2006.

[9] statsmodels development team. statsmodels. Python package, 2012.

[10] E. Suh. dirichlet 0.7. Python package, jan 2013.