Regression for Compositional Predictor Data

by

Zhenduo Huang

Master of Science (Mathematics), UNB, 2017

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

In the Graduate Academic Unit of Mathematics and Statistics

Supervisor(s):	Connie Stewart, PhD, Statistics
	Michael Tsagris, PhD, Statistics
Examining Board:	Matthew Stephenson, PhD, Statistics
	Jong-Kyou Kim, PhD, Computer Science

This report is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

October, 2022

Abstract

Compositional data refer to proportions of a whole. This type of data arise in many different disciplines, such as geology, biology, economics, and sociology, and require nontraditional methods for their analysis. In this report we consider regression methods for compositional data and introduce two new regression methods for compositional predictor data. The first method is unconstrained log-contrast (ULC) regression which is a less restrictive form log-contrast (LC) regression. The second proposed method is Greenacre's transformation regression which is an extension of ULC that allows for zeros in the compositional data. In this report we are interested in evaluating an F-test, in terms of its type I error rate and power, to compare LC regression with ULC regression, for various sample sizes and composition dimensions. Lastly, we used cross validation to assess the predictive performance of two new methods on three real-life datasets.

Table of Contents

Abstract ii

Table of Contents iii

List of Tables vi

List of Figures vii

1 Introduction 1

2 Compositional Data 4

2.1 Definitions 4

2.1.1 Composition 4

2.1.2 Closure 5

2.1.3 Subcomposition 6

2.1.4 Amalgamations 7

2.2 Zeros in Compositional Data 8

2.3 Transformations for Compositional Data 10

2.3.1 Additive Log-Ratio (ALR) Transformation 10

2.3.2 Centered Log-Ratio (CLR) and Isometric Log-Ratio (ILR) Transformations 11

2.3.3 α-transformation 12

2.3.4 Greenacre's Transformation 12

3 Regression for Compositional Data **14**

3.1 Regression for Compositional Response Data 15

3.1.1 Additive Log-Ratio (ALR) Regression 15

3.1.2 α-Regression 16

3.1.3 Dirichlet Regression 18

3.2 Regression for Compositional Predictor Data 21

3.2.1 Existing Regression Methods 21

3.2.2 New Regression Methods for Compositional Predictor Data . . 27

4 Simulation Study and Real-Life Data Analysis **31**

4.1 Simulation Study: LC vs ULC Regression 32

4.1.1 Type I Error Rate 33

4.1.2 Power 35

4.2 Simulation Study: Estimating the Power Transformation in Greenacre's Transformation Regression 39

4.3 Real-Life Data Analysis 42

4.3.1 The Mortality dataset 42

4.3.2 The lifeExpGdp dataset 43

4.3.3 The fgl dataset 44

5 Conclusion and Future Work **46**

Bibliography **48**

A Estimating the Power Transformation in Greenacre's Transformation Regression **51**

B Function for Simulation Study: LC vs ULC regression **52**

C Function for Simulation Study: Greenacre's Transformation Regression **55**

D The Distance Between the simulated λ and optimal λ **57**

Bibliography **51**

Vita

List of Tables

4.1 Proportion of simulation study iterations resulting in a Type I error in (a) and for which LC resulted in improved predictive accuracy in (b). 35

4.2 Power for different Discrepancies. 36

4.3 Proportion of times ULC is better for different Discrepancies. 37

List of Figures

4.1 *Disc* vs power with different n and D. 38

4.2 D vs power with different *Disc*. 39

4.3 n vs $\hat{\lambda} - \lambda$ for different combinations of λ and D. 41

4.4 D vs $\hat{\lambda} - \lambda$ for different combinations of λ and n. 42

Chapter 1

Introduction

Compositional data are positive multivariate data representing proportions of some whole whose vector elements, or components, sum to the same constant, which is usually taken to be 1 for convenience. These data arise in many different disciplines, such as geology, biology, economics, and sociology.

Regression analysis is a statistical technique which is used to estimate the relationship between some predictor variables and the response variable(s). Linear regression is the most common method in regression analysis; it requires that the residuals have a normal distribution, however, there are two restrictions for compositional response data: the first is that the sum of vector elements is equal to 1 and the second is that each element is greater than or equal to 0 but less than or equal to 1. Therefore, when the response variable is compositional data, the assumption of the residuals have a normal distribution is not valid, and when the predictor variables are compositonal data, the sum of the predictor variables are equal to 1 and may cause collinearity problems, since the predictor variables are linear dependent.

Various techniques exist for regression analysis when the response variables, predic-

tor variables or both are compositional. For compositional response data, a common approach is to transform the response data and then ordinary least squares regression can be used. Additive log-ratio (ALR) regression [14] uses the additive log-ratio transformation [1] while α-regression [10] uses the α-transformation [18] on the response variables. There also exists two regression models for compositional response data that assume a Dirichlet distribution. The first model was introduced in [3] but cannot accept zeros in the compositional data, while the second approach proposed in [11] is an extension that allows for zeros in the compositional data.

For regression with compositional predictor data, α-principal components regression [10] is a method that accepts zeros. Log-constrast (LC) regression is an alternative method which was first introduced in [2] and involves a log transformation of the predictor variables as well as a zero sum constraint on the coefficients. Since the predictor variables are log transformed, this method cannot accept zeros in the compositional data.

The above regression techniques are detailed in this report and two new regression methods for compositional predictor variables and a continuous response variable are proposed. The first new method is unconstrained log-contrast (ULC) regression, which is a less restrictive form of LC regression. Our second approach is Greenacre's transformation regression, which is an extension of ULC regression, using the Greenacre's transformation [12] and then ordinary least squares regression. In this report we are interested in evaluating an F-test, in terms of its type I error rate and power, to compare LC regression with ULC regression, for various sample sizes and composition dimensions. We also use a simulation study to investigate the bias in our estimate of the power transformation parameter which is a component of Greenacre's transformation.

Lastly, we analyze three real-life datasets with compositional predictor variables using our proposed methods. We use cross validation and the mean square prediction error to evaluate the techniques and compare them with some existing methods.

Chapter 2

Compositional Data

2.1 Definitions

2.1.1 Composition

In Chapter 1, we defined a composition as a non-negative vector $\mathbf{x}$ where all elements in the vector sum to 1. For example, the "ArcticLake" dataset in the "*robCompositions*" [20] package contains 39 sediment samples of sand, silt and clay at different depths of water (in meters) in an Arctic lake. The first sediment sample is at a depth equal to 10.4 meter with corresponding percentage of the sand, silt and clay components of (0.775, 0.195, 0.030). We will use $x_1, x_2, ..., x_D$ to denote the components of a D component composition $\mathbf{x}$.

A D-part composition must follow these two requirements:

1. Each component is non-negative and less than or equal to 1:

$$0 \leq x_j \leq 1, \quad \text{for } j = 1, ..., D. \tag{2.1}$$

2. The sum of all the components is 1:

$$x_1 + ... + x_D = 1. \tag{2.2}$$

Definition 2.1.1. *A composition* $\mathbf{x}$ *of* D *parts is a* $D \times 1$ *vector with non-negative components* $x_1, ..., x_D$ *whose sum is 1* [3].

Note that the composition can be specified by a d-part subvector such as $(x_1, ...x_d)$ where $d = D - 1$ since

$$x_D = 1 - x_1 - ... - x_d. \tag{2.3}$$

Definition 2.1.2. *The d-dimensional simplex embedded in the D-dimensional real space is the set defined by*

$$\mathbb{S}^d = \{(x_1, ..., x_D)^T | 0 \leq x_j \leq 1, \sum_{j=1}^{D} x_j = 1\}[3], \tag{2.4}$$

where $d = D - 1$.

The simplex is the sample space of D-dimensional compositional data.

2.1.2 Closure

Suppose $\mathbf{w}$ is a $D \times 1$ vector of non-negative components $w_1, ..., w_D$ with all measurements on the same scale. We often need to normalize the vector $\mathbf{w}$ in order for the components to sum to one. This operation is called closure and it is denoted by $\mathcal{C}(\cdot)$, where $(\cdot)$ denotes a vector. It is defined as:

$$\mathcal{C}(\mathbf{w}) = \left(\frac{w_1}{\sum_{j=1}^{D} w_j}, \frac{w_2}{\sum_{j=1}^{D} w_j}, \ldots, \frac{w_D}{\sum_{j=1}^{D} w_j} \right).$$

Definition 2.1.3. *Every vector* $\mathbf{w} \in \mathbb{R}_+^D$*, where* $\mathbb{R}_+^D$ *is the set of all positive real numbers, with a sum value* $t = w_1 + ... + w_D = \mathbf{j}'\mathbf{w}$ *can be normalized to a composition* $\mathbf{x}$*, where* $\mathbf{x} = \mathcal{C}(\mathbf{w}) = \mathbf{w}/t$ *and* $\mathbf{j}$ *is a* $D \times 1$ *vector with every entry being 1* [1].

For example, the closed composition for the vector (25,35,35) is (25/95,35/95,35/95) or (0.26,0.37,0.37).

2.1.3 Subcomposition

For a composition $\mathbf{x} \in \mathbb{S}^d$ there is often a subset of components that is of interest, and these form a subcomposition.

Definition 2.1.4. *If* S *is any subset of the parts* $1, ..., D$ *of a D-part composition* $\mathbf{x} \in \mathbb{S}^d$*, the subcomposition* $\mathcal{C}(\mathbf{x}_S) \in \mathbb{S}^{s-1}$*, where* s *is the number of components in the subset* S*, is the subvector formed from the corresponding components of* $\mathbf{x}$ [3].

We can use a selection matrix, defined below, to select components in a subcomposition.

Definition 2.1.5. *A selection matrix* $\boldsymbol{Q}$ *is a* $C \times D$ *matrix with* C *elements equal to 1,respectively to one in each row and at most one in each column, and with the remaining* $C \times (D-1)$ *elements equal to 0* [3].

Example 1. *In [3], there is a 5-part dataset named "Mineral compositions of 12 thin sections of granite". The second row of the data* $(\boldsymbol{x})$ *is* $Quartz = 0.228$, $Microcline = 0.269$, $Plagioclase = 0.420$, $Biotite = 0.064$ *and* $Others = 0.019$. *However,*

the Others are not of interest. Then the 4-part composition $\boldsymbol{y}$ *can be achieved by* $\boldsymbol{y} = \mathcal{C}(\boldsymbol{Q}\boldsymbol{x})$*, with the selection matrix*

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

2.1.4 Amalgamations

In some situations, it may be of interest to merge several components of a composition, resulting in a new composition of lower dimension.

Definition 2.1.6. *If the parts of a* D*-part composition are separated into* $C(\leq D)$ *mutually exclusive and exhaustive subsets and the components within each subset are added together, the resulting* C*-part composition is termed an amalgamation* [3].

Definition 2.1.7. *An amalgamating matrix* $\boldsymbol{A}$ *is a* $C \times D$ *matrix with* D *elements equal to 1, one in each column and at least one in each row, and with the remaining* $(C-1) \times D$ *elements equal to 0* [3].

Example 2. *If the second, third and fifth components in a 5-part composition* $\boldsymbol{x}$ *are to be amalgamated; then the 3-part composition* $\boldsymbol{y}$ *can be achieved by* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$*, with*

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

2.2 Zeros in Compositional Data

Many techniques for the analysis of compositional data assume components be strictly greater than zero, but in practice the data may contain zeros. In compositional data, zeros can be classified into two types [1]:

- Rounded zeros (trace zeros) are a result of the measurement process, where the observation has been recorded as zero when it is smaller than some value. The value may be the detection limit defined as the lowest signal, or the lowest corresponding quantity to be determined (or extracted) from the signal.
- Essential zeros (true or structural zeros) are recorded as zero as an indication of the complete absence of the component in the composition.

Example 3. *Suppose compositional data on expenses are recorded to two decimal places. For one observation, a person might spend \$2 for a coffee, pay \$100,000 to buy a new car and forget to buy a gift for his wife (spent \$0). Then the composition of money spent on this day, rounded to two decimal places would be:*

Item	*Coffee*	*Car*	*Gift*
Composition	*0.00*	*1.00*	*0.00*

In Example 3 the first zero, which is the proportion spent of coffee, is a rounded zero, and the second zero, which is the proportion spent of the gift, is an essential zero.

We now consider methods for handling the zeros in compositional data. In [3], it is suggested that amalgamation is one option to handle either type of zeros, if the components with zeros can be added to a related component. One disadvantage of amalgamation is that it will change the original compositional data and it may be amalgamating two or more useful components together. Another disadvantage is

that the compositional data may still contain zeros after the amalgamation.

A common approach for dealing with rounded zeros is to impute them with some small amount. In [5], a method using multiplicative replacement that maintains the ratios of nonzero components is proposed. For example, $\mathbf{x} = (x_1, x_2, ...x_D)$ be a composition that contains b zeros, then the elements of modified composition $\mathbf{y}$ are:

$$y_j = \begin{cases} \delta & \text{if } x_j = 0, \\ (1 - b\delta)x_j & \text{if } x_j \neq 0, \end{cases}$$

where δ is the replacement value for zeros and j goes from 1 to D. In [5], it is suggested that the best result is obtained when δ is set close to 65% of the detection limit.

Example 4. *Consider a 3-part composition,* $(0.67, 0.33, 0)$*, with one zero which has been recorded to two decimal places. Therefore, if the detection limit is 0.01, and we let* $\delta = 0.01 \times 0.65 = 0.0065$*, the modified composition* $\boldsymbol{w}$ *will be*
$\boldsymbol{w} = (0.67 \times (1 - 0.0065), 0.33 \times (1 - 0.0065), 0.0065) = (0.6656, 0.3279, 0.0065)$.

A disadvantage of this method is that all the zeros are replaced by the same number, no matter what the true values are, and the correlation between components which have zero values in the same composition is artificially set. There exists some other methods for dealing with rounded zeros such as the beta regression based strategy [14], or the Dirichlet regression method for compositional data can dealing with any type of zeros [11]. Methods for handling essential zeros in the regression context will be discussed in the next chapter.

There exists other methods for dealing with zeros such as Dirichlet regression method for compositional data with any type of zeros, or the beta regression based strategy with rounded zeros.

2.3 Transformations for Compositional Data

Recall that any D-part composition $\mathbf{x} = (x_1, ...x_D)$ belongs to $\mathbb{S}^d$. Compared to the sample space of the multivariate normal distribution (i.e. Euclidean space), the simplex is more restrictive since

1. $\sum_{j=1}^{D} x_j = 1$;
2. $0 \leq x_j \leq 1$ for $j = 1, ..., D$.

Several transformations exist for transforming the simplex to Euclidean space, and a common approach for analysing compositional data is to transform the data and then assume multivariate normality.

2.3.1 Additive Log-Ratio (ALR) Transformation

The ALR-transformation [1] is a one-to-one transformation that maps a D-part composition in $\mathbb{S}^d$ to a d-dimensional vector in $\mathbb{R}^d$ by using one component as the divisor (usually taken to be x_D) of all other components, where $d = D - 1$. It can be defined as

$$z_j = \log\left(\frac{x_j}{x_D}\right) \quad \text{for } j = 1, ...d. \tag{2.5}$$

The inverse of the ALR-transformation is

$$\begin{aligned} x_j &= \tfrac{e^{z_j}}{1+\sum_{k=1}^{d} e^{z_k}}, \quad \text{for } j = 1, ..., d \\ x_D &= \tfrac{1}{1+\sum_{k=1}^{d} e^{z_k}}. \end{aligned} \tag{2.6}$$

2.3.2 Centered Log-Ratio (CLR) and Isometric Log-Ratio (ILR) Transformations

Since the ALR-transformation uses the last component as the denominator, it is asymmetric and dependent on the choice of the value for the divisor. However, symmetry is an extremely desirable property for many types of analyses. In [2], the CLR-transformation is defined as

$$w_j = \log\left(\frac{x_j}{\prod_{k=1}^{D} x_k^{\frac{1}{D}}}\right), \quad \text{for } j = 1, ..., D. \tag{2.7}$$

Note that the denominator $\prod_{k=1}^{D} x_k^{\frac{1}{D}}$ is the geometric mean.

An obvious disadvantage of the CLR-transformation is that the sum of the transformed data is zero. In order to remove this restriction of the CLR-transformation, in [4], the isometric log-ratio (ILR) transformation is defined as

$$\mathbf{z} = \mathbf{w}\mathbf{H}^T, \tag{2.8}$$

where the components of $\mathbf{w}$ are given in Equation (2.7), and $\mathbf{H}$, which is a $d \times D$ orthonormal matrix, is the Helmert sub-matrix obtained by omitting the first row of the Helmert matrix [8].

Example 5. *When $D = 4$, the Helmert sub-matrix is*

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{12}} & -\frac{3}{\sqrt{12}} \end{bmatrix}.$$

2.3.3 α-transformation

From Equations (2.5), (2.7) and (2.8), we can see that all the above transformations are not applicable when there are zero values in the compositional data because the logarithm of zero is undefined. In [18], it is developed a more general and flexible transformation named the α-transformation, which allows for zeros in the compositional data. The α-transformation is carried out in two steps. First, let us define

$$\mathbf{w}_\alpha = \left(\frac{x_1^\alpha}{\sum_{j=1}^{D} x_j^\alpha}, ..., \frac{x_D^\alpha}{\sum_{j=1}^{D} x_j^\alpha} \right)^T , \tag{2.9}$$

which is referred to as the power transformation of a composition [3]. Second, define the α-transformation as

$$\mathbf{z}_\alpha = \frac{1}{\alpha} H \left(D\mathbf{w}_\alpha - \mathbf{j}_D \right), \tag{2.10}$$

where $\mathbf{j}_D$ is a vector with ones. Note that the α-transformation maps a D-part composition in $\mathbb{S}^d$ to a d-dimensional subset of $\mathbb{R}^d$. Also note that the α-transformation in Equation (2.10) converges to the ILR-transformation, which is the Equation (2.8), when α tends to zero. The value of α is usually chosen to be from -1 to 1, except when there are zeros in the composition; in which case α needs to be greater than zero.

2.3.4 Greenacre's Transformation

Greenacre's transformation [12], also called Greenacre's power transformation, is similar to the well-known Box-Cox transformation but with $-1 \leq \lambda \leq 1$. The

transformation is defined as

$$z_j = \begin{cases} \frac{x_j^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ ln(x_j) & \text{if } \lambda = 0. \end{cases} \tag{2.11}$$

Note that Greenacre's transformation in Equation (2.11) converges to the log-transformation as λ tends to zero. Also note that the value of the Greenacre's power, λ, has to be greater than zero if zero values exist in the compositional data.

Chapter 3

Regression for Compositional Data

Many regression models have been proposed for analyzing compositional data as the response data, predictor data or both. In this chapter, we review several well-known regression models for compositional response and predictor data. Also, we introduce two new regression models for compositional predictor data. In this chapter we use n for the sample size, D as the number of components in the compositional data.

Before we introduce the methods, we first need to introduce the well-known least squares estimator.

The method of least squares is an estimation method which estimates $\hat{\boldsymbol{\beta}}$ as those values with minimize the sum of squares errors, where the sum of squares error (SSE) is:

$$SSE(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

and where $\hat{y}_i$ is the fitted value. When there are linear relation between the response variable $\mathbf{Y}$ and the predictor variables $\mathbf{X}$, such as:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{E}, \tag{3.1}$$

where $\boldsymbol{\beta}$ is the vector (matrix) of regression coefficients, and $\mathbf{E}$ is the error terms. The least squares estimator $\hat{\boldsymbol{\beta}}_{ols}$ is given by

$$\hat{\boldsymbol{\beta}}_{ols} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}, \tag{3.2}$$

which is also called the classical multiple regression method.

3.1 Regression for Compositional Response Data

In this section, we will introduce three existing regression methods when the predictor variables follow the normal distribution with $\mathbf{X}$ as the $n \times (p+1)$ design matrix, which contains the p predictor variables along with the column of ones, and response variable $\mathbf{Y}$ is compositional data. Then, we use $\mathbf{Z}$ to denote the transformed response variables.

3.1.1 Additive Log-Ratio (ALR) Regression

ALR regression was introduced in [1] and uses the ALR transformed vector $\mathbf{z}_i = (z_{i1}, ..., z_{id})$, where

$$z_{ij} = log\left(\frac{y_{ij}}{y_{iD}}\right) \quad i = 1, ..., n, \quad j = 1, ..., d$$

as previously shown in Equation (2.5), where $\mathbf{y}_i = (y_{i1}, ..., y_{iD})$ denotes a compositional response vector. Then the least squares estimation can be used to get the coefficient $\hat{\boldsymbol{\beta}}_{ols}$ for the $n \times d$ transformed response matrix variables $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_n)^T$ and the predictor matrix $\mathbf{X}$. Note that the coefficients $\hat{\boldsymbol{\beta}}_{ols}$ is a $(p+1) \times d$ matrix.

ALR regression has the same drawback as the ALR transformation: it cannot accept any essential zero elements in the compositional response data (rounded zeros could

be imputed). Note that this regression method also can easily be extended for use with the CLR and ILR transformations (see Equations (2.7) and (2.8)), but they have the same drawback as ALR regression.

To carry out ALR regression in R we can use the function "*alr()*" from the package "*Compositional*" [19] to get the tranformed vectors. Then, we use "*lm()*" to get the coefficients $\boldsymbol{\beta}$ and the fitted values. Finally, we can use "*alrinv()*" from the package "*Compositional*" to transform the fitted values back to the simplex.

3.1.2 α-Regression

α-regression, which was introduced by [10], uses the α-transformed response vector $\mathbf{z}_\alpha$, where

$$\mathbf{z}_\alpha = \frac{1}{\alpha}\mathbf{H}\left(D\mathbf{w}_\alpha - \mathbf{j}_D\right),$$

with

$$\mathbf{w}_\alpha = \left(\frac{y_1^\alpha}{\sum_{j=1}^{D} y_j^\alpha}, ..., \frac{y_D^\alpha}{\sum_{j=1}^{D} y_j^\alpha}\right)^T$$

and $\mathbf{j}_D$ is a D vector with all ones. α-regression will use Equation (2.6), the inverse of the ALR transformation for the predicted response, to let the predicted response always lie within the simplex ($\mathbb{S}^d$). We can write the conditional mean of the observed composition as:

$$\mu_{ij} = \frac{e^{\mathbf{x}_i^T\boldsymbol{\beta}_j}}{1+\sum_{k=1}^{d} e^{\mathbf{x}_i^T\boldsymbol{\beta}_k}}, \quad \text{for} \quad j = 1, ..., d$$
$$\mu_{iD} = \frac{1}{1+\sum_{k=1}^{d} e^{\mathbf{x}_i^T\boldsymbol{\beta}_k}},$$

where $i = 1, ..., n$ and $\boldsymbol{\beta}_j = (\beta_{0j}, ..., \beta_{pj})^T$, $j = 1, ..., d$. The advantage of α-regression over ALR regression is that zeros are allowed and treated naturally as long as α is restricted to being greater than 0.

If we assume the transformed d-transformed data follow the multivariate normal distribution, parameter ($\hat{\boldsymbol{\Sigma}}_\alpha$) estimates are obtained by maximizing the log-likelihood function

$$l(\alpha) = -\frac{n}{2} log|\hat{\boldsymbol{\Sigma}}_\alpha| - \frac{1}{2} tr\left[(\mathbf{Z}_\alpha - \mathbf{M}_\alpha)\hat{\boldsymbol{\Sigma}}_\alpha^{-1}(\mathbf{Z}_\alpha - \mathbf{M}_\alpha)^T\right], \tag{3.3}$$

where $\mathbf{Z}_\alpha$ is the $n \times d$ α-transformed response matrix and $\mathbf{M}_\alpha$ is the matrix of α-transformed fitted value. The covariance matrix $\hat{\boldsymbol{\Sigma}}_\alpha$ need not be numerically estimated. The unbiased maximum likelihood estimator of $\hat{\boldsymbol{\Sigma}}_\alpha$ is

$$\hat{\boldsymbol{\Sigma}}_\alpha = (n - p - 1)^{-1}\mathbf{Z}_\alpha^T \mathbf{P} \mathbf{Z}_\alpha, \tag{3.4}$$

with

$$\mathbf{P} = \mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T,$$

where $\mathbf{X}$ is the design matrix. The coefficient estimates $\boldsymbol{\beta}$ for each α must be found using numerical optimization. Then, we can choose the optimal α by using the Kullback-Leibler divergence where

$$KL = 2\sum_{j=1}^{D}\sum_{i=1}^{n} y_{ij} log\frac{y_{ij}}{\hat{y}_{ij}}, \tag{3.5}$$

which is a measure of divergence between compositional vectors where $\hat{y}_{ij}$ denotes the fitted value. Hence, we transform the compositional data with different values of α, and then for every value of α we estimate the coefficients of $\boldsymbol{\beta}$ and the fitted value. Finally, we choose the value of α which minimizes Equation (3.5).

The α-transformation has two features:

1. When there are zeros in the compositional data, $0 < \alpha \leq 1$;

2. As α converges to 0, α-regression converges to ALR regression.

In *R* we can use the function "*alfa.reg()*" from the package "*Compositional*" to get the coefficients $\boldsymbol{\beta}$ and the fitted compositional values. In order to get the optimal α, we can use "*alfareg.tune()*", which measures the Kullback-Leibler divergence with a set of α from -1 to 1 (usually a sequence with distance equal to 0.1), to get the optimal α which has the lowest value of the Kullback-Leibler divergence in Equation (3.5).

3.1.3 Dirichlet Regression

The Dirichlet distribution is a family of distributions used to model continuous multivariate probabilities. It was used for regression analysis of compositional data in [6] and [7] for the cases in which covariates are present. If the composition $\mathbf{y}$ follows the Dirichlet distribution ($\mathbf{Y} \sim Dir(\phi, \mathbf{a}^*)$), the probability density function can be written as follows:

$$f(\mathbf{y}) = \frac{\Gamma\left(\sum_{j=1}^{D} \phi a_j^*\right)}{\prod_{j=1}^{D} \Gamma(\phi a_j^*)} \prod_{j=1}^{D} y_i^{(\phi a_j^* - 1)}, \tag{3.6}$$

where ϕ is the concentration parameter of a Dirichlet distribution. In the regression setting the parameters $a_j^* = E(y_j)$ with $\sum_{j=1}^{D} a_j^* = 1$ can be written as

$$\begin{aligned} a_j^* &= \frac{e^{\mathbf{X}^T \boldsymbol{\beta}_j}}{1+\sum_{k=1}^{d} e^{\mathbf{X}^T \boldsymbol{\beta_k}}}, \qquad j = 1, ..., d, \\ a_D^* &= \frac{1}{1+\sum_{k=1}^{d} e^{\mathbf{X}^T \boldsymbol{\beta_k}}}, \end{aligned} \tag{3.7}$$

where $\boldsymbol{\beta}_j = (\beta_{0j}, ..., \beta_{pj})^T$ is the matrix of coefficients. Note that Equation (3.7) is same as the inverse of the ALR-transformation in Equation (2.5). Given a random

sample of compositional data where $\mathbf{y}_1, \ldots, \mathbf{y}_n$ is assumed to be Dirichlet distributed, the log-likelihood function is

$$\ell = nlog\Gamma(\phi) - \sum_{j=1}^{D}\sum_{i=1}^{n} log\Gamma(\phi a_{ij}^*) + \sum_{j=1}^{D}\sum_{i=1}^{n}(\phi a_{ij}^* - 1)log(y_{ij}). \tag{3.8}$$

To carry out the Dirichlet regression in R, we can use the function *diri.reg()* from the package "*Compositional*".

From the log-likelihood equation we can see that the Dirichlet regression also cannot accept any zeros in the compositional response data since $log(y_{ij})$ does not exist when y_{ij}=0. In [11], an extension to the Dirichlet regression that allows for zeros of any type was suggested. Their method involves separating the original compositional observations $(\mathbf{y}_1, ..., \mathbf{y}_n)$ with zeros into B subgroups of compositional data S_b, where $b = 1, ..., B$. Every composition $\mathbf{y}_i,\ i = 1, ..., n$ only belongs to one subgroup based on where the zeros are located (the first subgroup S_1 contains all original compositional observations with no zeros). For example, all compositions with zeros in the first and second components will belong the same subgroup. In [11], it was shown that the Dirichlet log-likelihood function in Equation (3.8), adjusted to accept zero values, becomes

$$\begin{aligned}
\ell = \quad & n_1 log(\Gamma(\phi_1)) + n_1 log(\theta_1) - \sum_{i:y_i \in S_1}\sum_{j=1}^{D_1} log\Gamma(\phi_1 a_{1ij}^*) + \sum_{i:y_i \in S_1}\sum_{j=1}^{D_1}(\phi_1 a_{1ij}^* - 1)log(y_{1ij}) + ... \\
+ \quad & n_b log(\Gamma(\phi_b)) + n_b log(\theta_b) - \sum_{i:y_i \in S_b}\sum_{j=1}^{D_b} log\Gamma(\phi_b a_{bij}^*) + \sum_{i:y_i \in S_b}\sum_{j=1}^{D_b}(\phi_b a_{bij}^* - 1)log(y_{bij}) + ... \\
+ \quad & n_B log(\Gamma(\phi_B)) + n_B log(\theta_B) - \sum_{i:y_i \in S_B}\sum_{j=1}^{D_B} log\Gamma(\phi_B a_{Bij}^*) + \sum_{i:y_i \in S_B}\sum_{j=1}^{D_B}(\phi_B a_{Bij}^* - 1)log(y_{Bij}),
\end{aligned} \tag{3.9}$$

where D_b denotes the number of non-zero components in the subgroup of compositional data and n_b denotes the number of observations in the subgroup of compositional data S_b with $\sum_{b=1}^{B} n_b = n$. The a_{bj}^* are defined in Equation (3.7). The parameter θ_b is the probability that an observation comes from population b with $\sum_{b=1}^{B} \theta_b = 1$ and the maximum likelihood estimator of $\hat{\theta}_b$ is $\frac{n_b}{n}$. The difficulty with

maximizing the log-likelihood in Equation (3.9) is that there may be many groups with few observations, especially when the sample size is large. In this case, the parameter estimates found by maximizing Equation (3.9) are either not able to be obtained or may be unreliable. In [11], it used the selection matrix $\mathbf{Q}_b$ as in Definition (2.1.5), $b = 1, ..., B$, which can pick out the non-zero elements of the composition $\mathbf{Y}$ corresponding to group b. Now for each subgroup b, $\mathbf{Y}_b \sim Dir(\phi_b, \mathbf{Q}_b\mathbf{a}^*)$, where the parameter $\mathbf{Q}_b\mathbf{a}^*$ becomes a vector of length D_b. Let $\mathbf{Q}_b\mathbf{a}^*[j]$ denote the j^{th} component of this vector and if we set the $\phi = \phi_1 = ... = \phi_B$, the modified log-likelihood in Equation (3.9) becomes:

$$\begin{aligned} l = \quad & nlog\Gamma(\phi) \\ & +n_1 log(\theta_1) - \sum_{i:y_i \in S_1} \sum_{j=1}^{D_1} log\Gamma(\phi Q_1 a_i^*[j]) + \sum_{i:y_i \in S_1} \sum_{j=1}^{D_1} (\phi Q_1 a_i^*[j] - 1) log y_{1ij} + ... \\ & +n_B log(\theta_B) - \sum_{i:y_i \in S_B} \sum_{j=1}^{D_B} log\Gamma(\phi Q_B a_i^*[j]) + \sum_{i:y_i \in S_B} \sum_{j=1}^{D_B} (\phi Q_B a_i^*[j] - 1) log y_{Bij}. \end{aligned} \tag{3.10}$$

In order to maximize the log-likelihood, starting values are needed in the optimization algorithm, and we can use the zero-free subset of compositional data to get initial estimated coefficients as the least squared estimator

$$\boldsymbol{\beta_{ini}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Z}, \tag{3.11}$$

where $\mathbf{Z}$ is the $n_1 \times d$ ALR-transformed matrix by Equation (2.5) and $\mathbf{X}$ is the $n_1 \times (p+1)$ predictor matrix with n_1 denoting the size of the zero-free subset of original compositional observations. Then we use the same $\boldsymbol{\beta_{ini}}$ for all subgroups as the initial value in maximizing Equation (3.10) to obtain the optimal estimated coefficients.

In R we can use the function "*diri.reg()*" from package "*Compositional*" to get the

coefficients $\boldsymbol{\beta}$ and the fitted compositional values for the Dirichlet regression without zeros in the compositional response data, or use "*zadr()*" for the compositional response data with zeros.

The focus of this report is on regression for compositional predictor data; for more detailed information on regression for compositional response data, see [14] for ALR regression, [11] for Dirichlet regression and [10] for α-regression.

3.2 Regression for Compositional Predictor Data

In this section, we will introduce two existing and two new regression methods. We will denote the response variable, which is assumed to follow a normal distribution, as $\mathbf{Y}$. The predictor variable, the $n \times D$ compositional data, as $\mathbf{X}$. Then, $\mathbf{Z}$ is used to denote the transformed predictor variable.

3.2.1 Existing Regression Methods

α-PCR

Principal Component Analysis (PCA) is the process of computing the principal components (PCs), which are new variables that are constructed as linear combinations of the initial variables. PCs are uncorrelated and in PCA we use a reduced number of PCs to explain as much of the total variance as possible [10]. In Principal Component Regression (PCR), instead of regressing the dependent variables on the explanatory variables directly, the PCs of the explanatory variables are used as predictor variables. Often the first few PCs are sufficient for explaining a large proportion of the total variance allowing us to ignore the rest. PCR can be done in the following steps:

Step 1: Calculate the eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_p$ of the sample correlation matrix of the variate $\mathbf{X}$, where p denotes the number of predictor variables. These eigenvectors are associated with their corresponding eigenvalues $\lambda_1 \geq ... \geq \lambda_p \geq 0$ which are the variances of the corresponding PCs. The matrix of the eigenvectors is

$$\mathbf{V} = (\mathbf{v}_1, ..., \mathbf{v}_p)^T.$$

Step 2: Compute the scores ($\mathbf{S}$), which are linear combinations of the data that are determined by the coefficients for the first k PCs by

$$\mathbf{S} = \mathbf{X}\mathbf{V}_k,$$

note that $\mathbf{V}_k$ is a $p \times k$ matrix with columns corresponding to the first k eigenvectors in $\mathbf{V}$. $\mathbf{S}$ is the $n \times k$ matrix containing the first k PCs variables which will be used in the regression analysis in place of the original predictor data.

PCR combined with the α-transformation is a method named α-PCR which was introduced by [10] and can be used when the predictor variables are compositional.

α-PCR can be applied using the following steps:

1. Choose a value of α, apply the α-transformation in Equation (2.10) to the compositional predictor data $\mathbf{X}$ to obtain matrix $\mathbf{Z}_\alpha$.

2. Perform PCA on $\mathbf{Z}_\alpha$ to get the independent predictor variables $\mathbf{S}_\alpha$.

3. Estimate the regression coefficients by the least squares estimator

$$\hat{\boldsymbol{\beta}} = \mathbf{V}_k(\mathbf{S}_\alpha^T\mathbf{S}_\alpha)^{-1}\mathbf{S}_\alpha^T\mathbf{Y}. \tag{3.12}$$

Any number (k) of PCs can be used; if we use the first PC only, then $k = 1$, and if we use all PCs ($k = p$) then we end up with the same regression coefficients as if we used a classical multiple regression analysis.

α-PCR can use B-fold cross-validation based on the mean square prediction error (MSPE) to choose the optimal values of α and k. Specifically, the MSPE is the mean of the square of the difference between the fitted values and the observed values. In order to use cross-validation, we first need to split the transformed data into B folds randomly. Then, let each fold F_b, where $b = 1, ..., B$, be the test set, and all the other folds (fold number not b) as the training sets. Use α-PCR to estimate the coefficients $\hat{\boldsymbol{\beta}}_b$ by Equation (3.12) for each fold b. Lastly, the B-fold cross-validation MSPE (CV) can be measured by:

$$\begin{aligned} MSPE_b &= \tfrac{1}{n_b} \sum_{i:y_i \in F_b} (y_i - \hat{y}_i)^2, \quad i = 1, ..., n_b, \\ CV &= \tfrac{1}{B} \sum_{b=1}^{B} MSPE_b, \end{aligned} \tag{3.13}$$

where $\hat{y}_i = \mathbf{Z}_\alpha[i]\hat{\boldsymbol{\beta}}_b$ and $\mathbf{Z}_\alpha[i]$ is the i^{th} observation of α-transformed predictor data, and n_b is the sample size of fold b.

B-fold cross-validated score calculated with different sets of α and k, the set with the lowest B-fold cross-validated score is the optimal set of α and k.

The search values of α are usually chosen from -1 to 1 by increments of 0.1 unless there are zeros in the data, and k can be selected from 1 to p. In R we can use the function "*alfapcr.tune()*" from package "*Compositional*" to get the optimal set of α and k which have the lowest value of cross-validation MSPE in Equation (3.13). This function uses the function "*alfa.pcr()*" to get the cross-validation MSPE for a sequence of α and k. Then, we can use the function "*alfa.pcr()*" again for the opti-

mal set of α and k, in the whole sample to get the coefficients $\boldsymbol{\beta}$ and the fitted values.

Log-Contrast (LC) Regression

In log-contrast regression, the predictor variables are log transformed and a zero sum constraint is imposed on the coefficients. To motivate LC regression, recall the ALR transformation in Equation (2.5) and note that

$$\begin{aligned}\sum_{j=1}^{d} a_j log(\tfrac{x_j}{x_D}) &= \sum_{j=1}^{d} a_j(log(x_j) - log(x_D)) \\ &= \sum_{j=1}^{d} a_j(log(x_j)) - (\sum_{j=1}^{d} a_j)log(x_D) \\ &= \sum_{j=1}^{d} a_j log(x_j) + a_D log(x_D) \\ &= \sum_{j=1}^{D} a_j log(x_j),\end{aligned}$$

where $a_D = -\sum_{j=1}^{d} a_j$ and $\sum_{j=1}^{D} a_j = a_1 + ... + a_D = 0$. This log-linear combination is termed a log-contrast and is formally defined below.

Definition 3.2.1. *A log-contrast (LC) of a D-part composition $\boldsymbol{x}$ is any log-linear combination:*

$$a_1 log(x_1) + ... + a_D log(x_D) = \boldsymbol{a}' log(\boldsymbol{x})$$

with

$$a_1 + ... + a_D = \boldsymbol{a}' \boldsymbol{j}_D = 0,$$

where $\boldsymbol{a} = (a_1, ..., a_D)$ and $\boldsymbol{j}_D$ is a D dimensional vector with all ones. [3]

The LC has the following properties:

1. All LC regression models are permutation invariant. For example, if composi-

tion $\boldsymbol{x} = (x_1, ...x_d, x_D)$ is permuted to $(x_1, ...x_D, x_d)$, then

$$\begin{aligned} &a_1 log(x_1) + ... + a_d log(x_d) + a_D log(x_D) \\ =&a_1 log(x_1) + ... + a_D log(x_D) + a_d log(x_d). \end{aligned}$$

2. The LC regression model are scale-free in the sense that

$$\mathbf{a}' log(k\mathbf{x}) = \mathbf{a}' log\mathbf{x},$$

where $\mathbf{a} = (a_1, ...a_D)^T$. The proof is as follows:

$$\begin{aligned} \mathbf{a}' log(k\mathbf{x}) =&a_1 log(kx_1) + ... + a_d log(kx_d) + a_D log(kx_D) \\ =&a_1 log(kx_1) + ... + a_d log(kx_d) + (-(a_1 + ... + a_d)) log(kx_D) \\ =&a_1 log\left(\frac{kx_1}{kx_D}\right) + ... + a_d log\left(\frac{kx_d}{kx_D}\right) \\ =&\mathbf{a}' log\mathbf{x}. \end{aligned}$$

Note that the CLR transformation is also a log-contrast. CLR uses the geometric mean ($g(\mathbf{x}) = \prod_{j=1}^{D} x_j^{\frac{1}{D}}$) as the divisor and since $g(\mathbf{x})$ is a constant, it can be shown that the CLR transformation is a log-contrast by setting $k = 1/(g(\mathbf{x}))$ in the proof of Property 2. The properties point out the coefficient will not change, no matter the order of the component, and since the a_0 will not change by the scale, therefore, we can set it equal to zero for the simulation study in the next chapter to make it easy to study.

LC regression involves relating response data to predictor data using the LC relationship and estimating the coefficients subject to the constraint that the sum of the coefficients is zero. That is, LC regression assumes the relationship

$$y = \beta_0 + \beta_1 log(x_1) + ... + \beta_d log(x_d) + \beta_D log(x_D) + e \tag{3.14}$$

with

$$\sum_{j=1}^{D} \beta_j = 0, \tag{3.15}$$

where $e \sim N(0, \sigma^2)$ is the error term.

The constrained least squares estimator can be denoted as $\widetilde{\boldsymbol{\beta}}_{cls}$ and is given by

$$\widetilde{\boldsymbol{\beta}}_{cls} = \underset{\boldsymbol{\beta}'\boldsymbol{j}=0}{argmin}\, SSE(\boldsymbol{\beta}) \tag{3.16}$$

where

$$SSE(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_j - log(\mathbf{x}_i')\boldsymbol{\beta})^2.$$

The constrained least squares estimator $\widetilde{\boldsymbol{\beta}}_{cls}$ minimizes the SSE over all $\boldsymbol{\beta} \in \mathbb{R}^D$ subject to the sum of the coefficients being zero. Since $\widetilde{\boldsymbol{\beta}}_{cls}$ is a restricted estimator, we use a tilde " $\sim$ " instead of the conventional " $\wedge$ " in order to denote that the estimation method is constrained least squares. Since there are constraints for the coefficients, we can use the technique of Lagrange multipliers ($\mathcal{L}$) to minimize the Equation (3.16) as:

$$\mathcal{L}(\widetilde{\boldsymbol{\beta}}_{cls}, \boldsymbol{\lambda}) = \frac{1}{2} SSE(\widetilde{\boldsymbol{\beta}}_{cls}) + \boldsymbol{\lambda}'\boldsymbol{j}'\widetilde{\boldsymbol{\beta}}_{cls}. \tag{3.17}$$

The solution point $(\widetilde{\boldsymbol{\beta}}_{cls}, \widetilde{\boldsymbol{\lambda}}_{cls})$ with the first-order conditions for $\mathcal{L}$ is:

$$\frac{\partial}{\partial \boldsymbol{\beta}} \mathcal{L}(\widetilde{\boldsymbol{\beta}}_{cls}, \widetilde{\boldsymbol{\lambda}}_{cls}) = -log(\mathbf{X})'\mathbf{Y} + log(\mathbf{X})'log(\mathbf{X})\widetilde{\boldsymbol{\beta}}_{cls} + \mathbf{R}\widetilde{\boldsymbol{\lambda}}_{cls} = 0 \tag{3.18}$$

and

$$\frac{\partial}{\partial \boldsymbol{\lambda}} \mathcal{L}(\widetilde{\boldsymbol{\beta}}_{cls}, \widetilde{\boldsymbol{\lambda}}_{cls}) = \mathbf{R}'\widetilde{\boldsymbol{\beta}}_{cls} = 0, \tag{3.19}$$

where $\mathbf{R} = (0, \mathbf{1}_D)^T$. If we pre-multiply Equation (3.18) by $\mathbf{R}'(log(\mathbf{X})'log(\mathbf{X}))^{-1}$, then Equation (3.18) becomes

$$-\mathbf{R}'\hat{\boldsymbol{\beta}}_{ols} + \mathbf{R}'\widetilde{\boldsymbol{\beta}}_{cls} + \mathbf{R}'(log(\mathbf{X})'log(\mathbf{X}))^{-1}\mathbf{R}\widetilde{\boldsymbol{\lambda}}_{cls} = 0, \tag{3.20}$$

where $\hat{\boldsymbol{\beta}}_{ols}$ is the ordinary least squares estimator. From Equation (3.19) we also have that $\mathbf{R}'\widetilde{\boldsymbol{\beta}}_{cls} = 0$. Therefore, the $\widetilde{\boldsymbol{\lambda}}_{cls}$ in Equation (3.20) will become

$$\widetilde{\boldsymbol{\lambda}}_{cls} = \left[\mathbf{R}'(log(\mathbf{X})'log(\mathbf{X}))^{-1}\mathbf{R}\right]^{-1}(\mathbf{R}'\hat{\boldsymbol{\beta}}_{ols}). \tag{3.21}$$

With Equation (3.21), the $\widetilde{\boldsymbol{\beta}}_{cls}$ in Equation (3.18) can be computed as,

$$\widetilde{\boldsymbol{\beta}}_{cls} = \hat{\boldsymbol{\beta}}_{ols} - \mathbf{GR}[\mathbf{R}'\mathbf{GR}]^{-1}\mathbf{R}'\hat{\boldsymbol{\beta}}_{ols}, \tag{3.22}$$

where $\mathbf{G} = (log(\mathbf{X})'log(\mathbf{X}))^{-1}$.

In *R* we can use the function "*lc.reg()*" from package "*Compositional*" to get the coefficients $\widetilde{\boldsymbol{\beta}}_{cls}$.

3.2.2 New Regression Methods for Compositional Predictor Data

We investigated two new alternative approaches to regression for compositional predictor data, namely unconstrained log-contrast (ULC) regression and an extension using the Greenacre's transformation discussed in Chapter 2.

ULC Regression

LC regression involves the constraint that the sum of the regression coefficients is

equal to zero. In the following we will introduce a log-transformed regression method without this constraint, and in Chapter 4 we will compare LC and ULC regression to determine whether the constraint is necessary.

We refer to the log-linear combination

$$y = a_1 log(x_1) + ... + a_D log(x_D), \tag{3.23}$$

without constraint on the coefficients as an unconstrained log-contrast (ULC). The ULC has the following properties:

1. ULC regression is permutation invariant. For example, if composition $\mathbf{x} = (x_1, ...x_d, x_D)$ is permuted to $(x_1, ...x_D, x_d)$, then

$$\begin{aligned} &a_1 log(x_1) + ... + a_d log(x_d) + a_D log(x_D) \\ =&a_1 log(x_1) + ... + a_D log(x_D) + a_d log(x_d). \end{aligned}$$

2. The coefficients (a_1,...,a_D) in the ULC regression are scale free in the sense that

$$a_0 + \mathbf{a}' log(k\mathbf{x}) = a_0^* + \mathbf{a}' log\mathbf{x}.$$

where $a_0^* = a_0 + (a_1 + ... + a_D) log(k)$. This property can be proved as

$$\begin{aligned} y =&a_0 + \mathbf{a}' log(k\mathbf{x}) \\ =&a_0 + a_1 log(kx_1) + ... + a_d log(kx_d) + a_D log(kx_D) \\ =&a_0 + a_1 log(x_1) + ... + a_D log(x_D) + (a_1 + ... + a_D) log(k) \\ =&\mathbf{a}' log(\mathbf{x}) + (a_0 + (a_1 + ... + a_D) log(k)) \\ =&a_0^* + \mathbf{a}' log(\mathbf{x}). \end{aligned}$$

As in the case of LC regression, ULC regression involves relating response data to predictor data but using the ULC relationship, which reduces to the method of multiple linear regression.

Compared to LC regression, ULC regression is simpler as it does not involve a constraint on the sum of the regression coefficients. Therefore, the unconstrained least squares (ULS) estimator, $\hat{\boldsymbol{\beta}}_{uls}$, is

$$\hat{\boldsymbol{\beta}}_{uls} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}, \tag{3.24}$$

where $\mathbf{Z} = log(\mathbf{X})$.

LC regression can be thought of as a reduced model of ULC regression, since the $\boldsymbol{\beta}_{uls}$ can be any set from $\mathbb{R}^D$. In the next chapter, we will evaluate the ability of an F-test to test if the the simpler model (LC) is adequate against the alternative hypothesis that the full model (ULC) is necessary.

In *R* we can use the function "*ulc.reg()*" from the package "*Compositional*" to get the coefficients $\hat{\boldsymbol{\beta}}_{uls}$.

Greenacre's Transformation Regression

Greenacre's transformation regression uses the Greenacre's transformed vector $\mathbf{z} = z_1, ..., z_D$, where recall that

$$z_j = \begin{cases} \frac{x_j^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ ln(x_j) & \text{if } \lambda = 0. \end{cases}$$

The relationship between the response variable $\mathbf{Y}$ and the transformed predictor

variables $\mathbf{Z}$ is

$$\mathbf{Y} = \mathbf{Z}'\boldsymbol{\beta}_\lambda + \mathbf{E}, \tag{3.25}$$

where, for a given value of λ, response vector $\mathbf{Y}$ and $n \times p$ matrix of transformed predictor variables $\mathbf{Z}$ and the regression coefficient estimator, $\hat{\boldsymbol{\beta}}_\lambda$ is the least squares estimator,

$$\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{Y}. \tag{3.26}$$

The λ is allowed to range from -1 to 1. The estimated regression coefficient $\hat{\boldsymbol{\beta}}_\lambda$ has an associated sum of squares error (SSE) for each λ. The ordinary least squares estimator $\hat{\boldsymbol{\beta}}$ is the one with the lowest SSE, and the parameter λ is estimated by the value of λ that minimizes the SSE. The following equation can determine the optimal λ:

$$\hat{\boldsymbol{\beta}} = \underset{-1\leq\lambda\leq 1}{argmin}\, SSE(\hat{\boldsymbol{\beta}}_\lambda), \tag{3.27}$$

where

$$SSE(\hat{\boldsymbol{\beta}}_\lambda) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

with $\hat{y}_i = \boldsymbol{z}_i'\hat{\boldsymbol{\beta}}_\lambda$, where $i = 1, ..., n$ indexing the observations.

In *R*, the function "*bculcreg.mse.tune*" in Appendix A can be optimized to find the lowest SSE and the best λ. In the next chapter we will assess the bias of our estimator of λ.

Chapter 4

Simulation Study and Real-Life Data Analysis

Type I and type II errors are two types of incorrect decisions. The probability of a type I error is the probability of rejecting a true null hypothesis. The probability of a type II error is the probability of not rejecting a false null hypothesis. Power is the probability of not making a type II error [22].

In this chapter, we will use simulated data and evaluate an ANOVA F-test, in terms of its type I error rate, to compare LC regression with ULC regression for various sample sizes and composition dimensions. Also, we will compare the cross-validated LC and ULC regression by computing the proportion of iterations for which the ULC regression had a lower prediction error than the LC regression. Next, we will evaluate an ANOVA F-test, in terms of its power, by letting the sum of the coefficients deviate incrementally from zero. Similarly, we will compare how the cross-validated LC and ULC regression methods change when the sum of the coefficients gets further away from 0. Then, for Greenacre's transformation regression, we will use the simulated data to study the bias in our estimator of λ.

In order to assess the usefulness of the new regression methods (ULC regression and Greenacre's transformation regression) in practice, we will consider three real-life datasets with which we will compare these two models to the LC regression method.

4.1 Simulation Study: LC vs ULC Regression

In this section we will compare LC regression with ULC regression for various sample sizes and composition dimensions. Recall that the difference between LC and ULC regression is the constraint that the sum of the regression coefficients equals zero for LC regression. We are interested in evaluating an F-test, in terms of its type I error rate and power for the following hypotheses:

$$\begin{aligned} H_0 &: \sum_{j=1}^{D} \beta_j = 0 \\ H_1 &: \sum_{j=1}^{D} \beta_j \neq 0. \end{aligned}$$

In Chapter 3 we pointed out that LC regression can be thought of as a reduced model of ULC regression, so we propose using the ANOVA F-test to check if the reduced model (LC regression) is adequate or not. The F-statistic can be measured as follows:

$$F^* = \frac{SSE(LC) - SSE(ULC)}{\frac{SSE(ULC)}{n-D}},$$

where n is the sample size and D is the number of $\boldsymbol{\beta}$ coefficients. The p-value is determined by comparing F^* to the $F_{1,n-D-1}$ (since the degrees of freedom of LC regression is $D-1$ and the degrees of freedom of ULC regression is D). Large values of F^* suggest that the sum of coefficients is not equal to zero (ULC is necessary).

4.1.1 Type I Error Rate

A type I error occurs if a test incorrectly rejects the null hypothesis. To estimate the probability of making a type I error using our proposed F-test, we first simulated data under H_0, with $\sum_{j=1}^{D} \beta_j = 0$ and applied the F-test. Then we computed the proportion of times we incorrectly rejected H_0 and compared this to the nominal significance level, $\alpha = 0.05$. More specifically, the data were simulated using the following steps:

Step 1: Set nine different sample sizes ($n = 50,\ 100,\ 200,\ 300,\ 500,\ 1000,\ 2000,\ 5000,\ 10000$) and set four different composition sizes ($D = 3, 5, 7, 10$).

Step 2: With n as the sample size and $\frac{1}{D}$ as the mean value for each components, we used the function *rdiri()* in the package "*Compositional*" to build a matrix of compositional predictors ($\mathbf{X}$) of dimension $n \times D$ and then log-transformed the matrix $\mathbf{X}$.

Step 3: Build a $D \times 1$ coefficient vector ($\boldsymbol{\beta}$) and ensure that the entries sum to zero. In this simulation study, we generated the first set of $\lceil \frac{D-1}{2} \rceil$ coefficients from a uniform distribution with parameters 1 and 2, the second set of $\lfloor \frac{D-1}{2} \rfloor$ coefficients from a uniform distribution with parameters -2 and -1, and the last coefficient equal to the negative of the sum of the previous D coefficients. Then, these coefficients are multiplied by the log-transformed compositional data ($log(\mathbf{X})$) to get $\mathbf{Y}_o$. Finally, add Gaussian noise to the $\mathbf{Y}_o$ to obtain the response variable ($\mathbf{Y}$), where we generated the noise from a normal distribution with a mean value is 0 and the standard deviation of $\mathbf{Y}_o$.

Note that for the simulated data, we set the intercept as 0, since the coefficient estimator $\boldsymbol{\beta}$ will not be changed when there have intercept value or not.

The above three steps yield simulated compositional predictor data $\mathbf{X}$ and corresponding response data $\mathbf{Y}$ with coefficients generated with the constraint. The following steps will be used to compare LC and ULC regression by estimating the probability of making type I error and computing the proportion of times cross validated of LC regression is lower than that of ULC regression.

Step 4: Apply *lc.reg()* and *ulc.reg()* functions from the package *Compositional* [19] to the simulated data to estimate the regression coefficients, and then carry out the F-test using *lcreg.aov()*, to decide whether the LC regression model is adequate. Store the p-value of the F-test and then measure the cross-validated for both models by Equation (3.13).

Step 5: For all combinations of n and D, repeat the above steps (1-4) 1000 times and then estimate the probability of making a type I error as the proportion of times the p-value from the F-test is less than $\alpha = 0.05$. Also, compute the proportion of times the cross-validation MSPE of LC regression is less than the cross-validation MSPE of ULC regression. The model that has the lower cross-validation MSPE is indicative of a better model.

The R code used to carry out this simulation study can be found in Appendix B.

Table 4.1(a) shows that the probability of making type I error is about 0.05 in all cases; therefore, the F-test performs well in terms of yielding an accurate probability

Table 4.1: Proportion of simulation study iterations resulting in a Type I error in (a) and for which LC resulted in improved predictive accuracy in (b).

(a) Proportion of Times Make Type I Error

n	D=3	D=5	D=7	D=10	Mean
50	0.053	0.044	0.046	0.051	0.048
100	0.046	0.052	0.045	0.055	0.050
200	0.047	0.059	0.060	0.039	0.051
300	0.057	0.042	0.049	0.055	0.051
500	0.052	0.049	0.038	0.048	0.047
1000	0.047	0.067	0.054	0.041	0.052
2000	0.069	0.061	0.053	0.059	0.060
5000	0.045	0.057	0.047	0.045	0.048
10000	0.052	0.062	0.054	0.060	0.057
Mean	0.052	0.055	0.050	0.050	0.052

(b) Proportion of Times LC Better

n	D=3	D=5	D=7	D=10	Mean
50	0.82	0.84	0.81	0.81	0.82
100	0.80	0.83	0.83	0.82	0.82
200	0.83	0.81	0.82	0.84	0.83
300	0.82	0.84	0.82	0.83	0.83
500	0.83	0.82	0.81	0.80	0.81
1000	0.82	0.80	0.80	0.83	0.81
2000	0.81	0.81	0.79	0.83	0.81
5000	0.83	0.82	0.80	0.82	0.82
10000	0.80	0.80	0.81	0.82	0.81
Mean	0.82	0.82	0.81	0.82	0.82

of type I error for this test.

Table 4.1(b) shows the Proportion of times LC regression have lower cross-validation MSPE than ULC regression. Based on the simulation experiment, the probability of LC regression being better in terms of predictive performance is about 82%. These percentages were fairly consistent across the sample sizes and components. Given that the correct model, in this case, is the LC model. It is perhaps surprising that almost 20% of the time, the ULC has a lower prediction error.

4.1.2 Power

Power is the probability of making a correct decision to reject the H_0 when the H_0 is false. To simulate data under H_1, we followed the steps in Subsection 4.1.1, except that in Step 3, after we get the beta coefficients ($\boldsymbol{\beta}$), we add a discrepancy value to each element in $\boldsymbol{\beta}$. Therefore, the sum of the $\boldsymbol{\beta}$ coefficients was equal to the number of components (D) times a non-zero $Disc$:

$$\sum_{j=1}^{D} \beta_j = Disc \times D,$$

where $Disc \neq 0$. Note that when the data generated in this way, they are generated under H_1. The discrepancy was set to several different possible values ($Disc = 0.1, 0.5, 1, 2$) in order to see how the power changes as the sum of the regression coefficients gets further from 0, as well as compare the predictive accuracy of the two models.

Table 4.2: Power for different Discrepancies.

(a) Power for $Disc = 0.1$

	D=3	D=5	D=7	D=10	Mean
50	0.059	0.047	0.054	0.049	0.052
100	0.051	0.052	0.058	0.061	0.056
200	0.044	0.055	0.045	0.057	0.050
300	0.058	0.058	0.051	0.059	0.056
500	0.050	0.039	0.064	0.057	0.052
1000	0.058	0.052	0.044	0.039	0.048
2000	0.048	0.054	0.064	0.063	0.057
5000	0.081	0.087	0.073	0.066	0.077
10000	0.116	0.094	0.109	0.090	0.102
Mean	0.063	0.060	0.062	0.060	0.061

(b) Power for $Disc = 0.5$

	D=3	D=5	D=7	D=10	Mean
50	0.064	0.058	0.057	0.049	0.057
100	0.061	0.064	0.050	0.059	0.058
200	0.079	0.068	0.062	0.075	0.071
300	0.115	0.088	0.091	0.075	0.092
500	0.125	0.108	0.100	0.094	0.107
1000	0.208	0.183	0.185	0.163	0.185
2000	0.386	0.312	0.322	0.220	0.310
5000	0.720	0.631	0.640	0.571	0.640
10000	0.929	0.905	0.890	0.849	0.893
Mean	0.299	0.269	0.266	0.239	0.268

(c) Power for $Disc = 1$

	D=3	D=5	D=7	D=10	Mean
50	0.065	0.068	0.071	0.059	0.066
100	0.129	0.108	0.113	0.077	0.107
200	0.163	0.156	0.143	0.145	0.152
300	0.241	0.221	0.203	0.176	0.210
500	0.330	0.286	0.301	0.284	0.300
1000	0.636	0.546	0.525	0.504	0.553
2000	0.867	0.838	0.801	0.736	0.810
5000	0.997	0.997	0.992	0.978	0.991
10000	1.000	1.000	1.000	1.000	1.000
Mean	0.492	0.469	0.461	0.440	0.465

(d) Power for $Disc = 2$

	D=3	D=5	D=7	D=10	Mean
50	0.156	0.157	0.139	0.111	0.141
100	0.337	0.240	0.213	0.188	0.244
200	0.500	0.449	0.420	0.370	0.435
300	0.660	0.597	0.595	0.553	0.601
500	0.853	0.813	0.805	0.724	0.799
1000	0.984	0.972	0.959	0.966	0.970
2000	1.000	1.000	1.000	1.000	1.000
5000	1.000	1.000	1.000	1.000	1.000
10000	1.000	1.000	1.000	1.000	1.000
Mean	0.721	0.692	0.681	0.657	0.688

When $Disc = 0.1$ (see Table 4.2(a)), the power was similar to the Type I error rates found in Table 4.1(a) where, in most cases, it was around 0.05. When the sample size was 10000; the power was slightly larger (around 0.10). Also, Table 4.3(a) shows the proportion of times that ULC offered better predictive accuracy. Compared to Table 4.1(b), most scenarios had a similar result; that is the proportion of times ULC regression was found to offer better of predictive performance was about 18%,

Table 4.3: Proportion of times ULC is better for different Discrepancies.

(a) $Disc = 0.1$

	D=3	D=5	D=7	D=10	Mean
50	0.19	0.19	0.20	0.18	0.19
100	0.17	0.19	0.19	0.20	0.19
200	0.17	0.18	0.18	0.18	0.18
300	0.19	0.19	0.19	0.20	0.19
500	0.17	0.16	0.21	0.19	0.18
1000	0.19	0.19	0.18	0.18	0.19
2000	0.21	0.18	0.20	0.19	0.20
5000	0.20	0.21	0.23	0.21	0.21
10000	0.30	0.25	0.26	0.23	0.26
Mean	0.20	0.19	0.21	0.19	0.20

(b) $Disc = 0.5$

	D=3	D=5	D=7	D=10	Mean
50	0.20	0.19	0.22	0.20	0.20
100	0.21	0.20	0.18	0.20	0.20
200	0.24	0.19	0.21	0.22	0.21
300	0.29	0.24	0.23	0.22	0.25
500	0.31	0.27	0.29	0.26	0.28
1000	0.41	0.36	0.37	0.35	0.37
2000	0.57	0.52	0.52	0.44	0.51
5000	0.85	0.79	0.80	0.75	0.80
10000	0.97	0.96	0.96	0.93	0.95
Mean	0.45	0.42	0.42	0.40	0.42

(c) $Disc = 1$

	D=3	D=5	D=7	D=10	Mean
50	0.21	0.20	0.24	0.19	0.21
100	0.31	0.26	0.28	0.24	0.27
200	0.36	0.35	0.33	0.34	0.35
300	0.44	0.41	0.39	0.40	0.41
500	0.57	0.48	0.51	0.49	0.51
1000	0.78	0.72	0.74	0.70	0.73
2000	0.94	0.94	0.91	0.87	0.91
5000	1.00	1.00	1.00	0.99	1.00
10000	1.00	1.00	1.00	1.00	1.00
Mean	0.62	0.60	0.60	0.58	0.60

(d) $Disc = 2$

	D=3	D=5	D=7	D=10	Mean
50	0.35	0.33	0.33	0.31	0.33
100	0.54	0.45	0.43	0.40	0.45
200	0.69	0.65	0.62	0.57	0.63
300	0.82	0.76	0.76	0.74	0.77
500	0.93	0.91	0.91	0.88	0.91
1000	0.99	0.99	0.98	0.99	0.99
2000	1.00	1.00	1.00	1.00	1.00
5000	1.00	1.00	1.00	1.00	1.00
10000	1.00	1.00	1.00	1.00	1.00
Mean	0.81	0.79	0.78	0.76	0.79

except when the sample size was 10000. In this case, ULC regression offered better predictive performance around 26% of all. Therefore, we can reasonably postulate that if the discrepancy is small, the F-test is not very powerful and also that the LC model may be preferred in terms of predictive performance over the ULC model.

From Table 4.2(b), when $Disc = 0.5$, the power, for all component sizes, was about 0.6 or more when $n \geq 5000$. For smaller sample sizes, the power was lower but most of the time it was greater than 0.05. Also Table 4.3(b) shows the proportion of times ULC offered better predictive accuracy than LC and this was at least 0.75 for the large sample size scenarios ($n \geq 5000$). When the sample size is 2000, ULC offered better predictive performance was approximately 50% of all. Table 4.2(b) also shows that for a given sample size, the power decreases slightly as D increases.

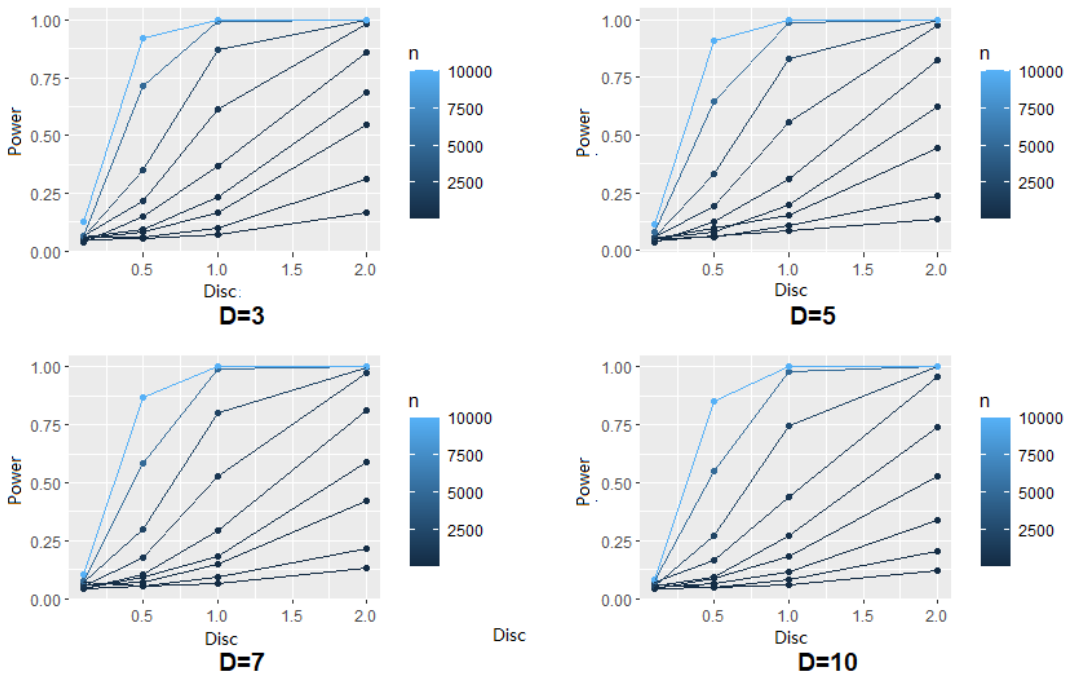

Figure 4.1: *Disc* vs power with different n and D.

When *Disc* was increased to 1, Table 4.2 (c) suggests that the power was greater than 0.6 when the sample size is larger than 1000, and from Table 4.3 (c), the mean proportion of the time ULC offered better predictive accuracy than LC was at least 0.7, so it was more likely (for $n \geq 1000$) that ULC will perform better.

Lastly, when we increased the *Disc* to 2, Table 4.2(d) shows that most of the time the power was greater than 0.6 except when the sample size was very small (such as $n = 50, 100$). Similarly, Table 4.3(d) shows that in almost all scenarios the ULC regression had a lower cross-validation MSPE than the LC method. Even at the smallest sample size (n=50), the probability that ULC offered better predictive accuracy was at least 0.31.

Figure 4.1 shows that the relationship between the discrepancy and the power curve

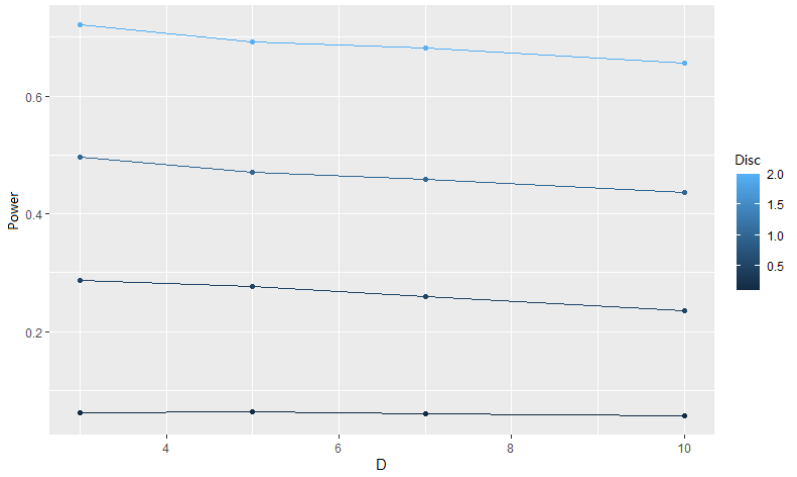

Figure 4.2: D vs power with different $Disc$.

approximately resembles that of the logistic function (that is an S-curve). The S-curve can be seen more clearly when the sample size was 1000 (the fourth line from top in each plot) or higher. Also, it can be seen from Figure 4.2, with D as the x-axis and the power average over all sample sizes for each D as the y-axis, that when D increases, the power decreases. However, the decreasing rates are small.

4.2 Simulation Study: Estimating the Power Transformation in Greenacre's Transformation Regression

In Chapter 3, we discussed a procedure for estimating λ that involved minimizing the SSE. In this section, we will use a simulation study to examine how accurately λ can be estimated using this procedure. In this simulation study we limited the λ to be positive in case we may have some zeros in the compositional data.

We simulated samples of data using the following steps:

Step 1: Set six different sample sizes (n = 100, 300, 500, 1000, 5000, 10000), four different numbers of components ($D = 3, 5, 7, 10$) and six different transformation values we choose were ($\lambda = 0,\ 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9$).

Step 2: Use the Dirichlet distribution to build a matrix of compositional predictors ($\mathbf{X}$) of dimension $n \times D + 1$, which contains the D predictor variables along with the column of ones, and then use Equation (2.11) to get Greenacre's transformed matrix $\mathbf{Z}$ for each sample size (n), number of components (D) and transformation value (λ).

Step 3: Build a $D+1$ vector of coefficients ($\boldsymbol{\beta}$) where one coefficient is the intercept, with some being positive values and the remaining being negative values. More specifically, we let the the first constant coefficient be 1; among the others, half of them were generated from a uniform distribution with parameters 1 and 2 and the remaining half are generated from a uniform distribution with parameters -2 and -1. The coefficients were then multiplied by the Greenacre's transformed matrix $\mathbf{Z}$ to generate $\mathbf{Y}_o$ and Gaussian noise was added to obtain the response variable ($\mathbf{Y}$), where noise was generated from a normal distribution with mean value 0 and the standard deviation of $\mathbf{Y}_o$.

The above three steps yields simulated compositional predictor data $\mathbf{X}$ and corresponding response data $\mathbf{Y}$. The following steps will be used to compute the $\hat{\lambda}$ and subsequently the difference between $\hat{\lambda}$ and the true λ.

Step 4: Apply the function *bculcreg.mse.tune()* In *R* (see Appendix A) package to the simulated data to estimate the $\hat{\lambda}$ and compute the difference from the $\hat{\lambda}$

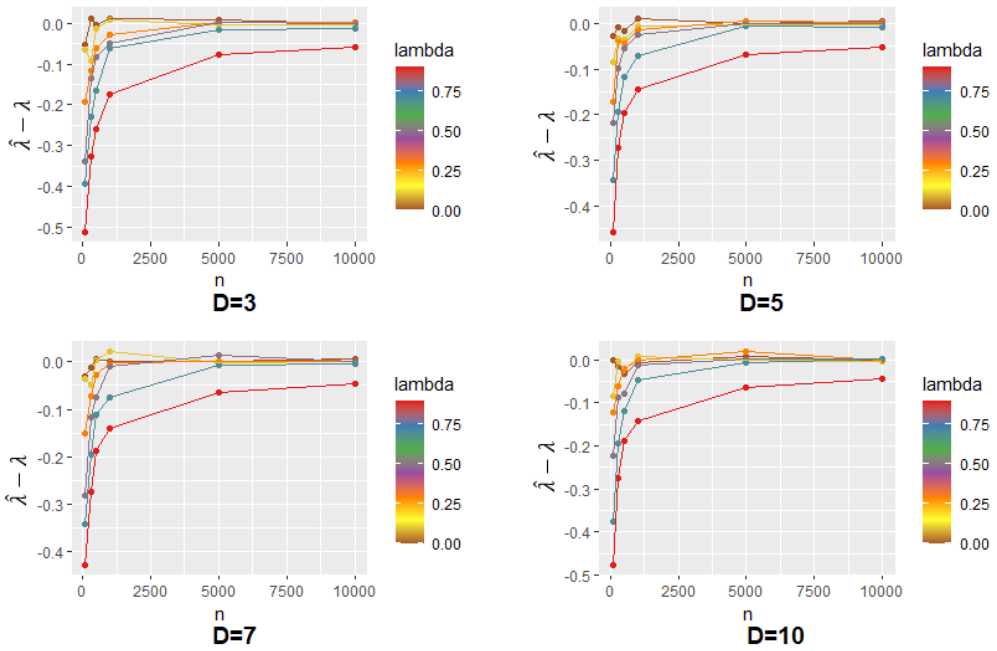

Figure 4.3: n vs $\hat{\lambda} - \lambda$ for different combinations of λ and D.

to the true λ.

Step 5: For all combinations of sample sizes, components and transformation values, repeat the above steps (1-4) 1000 times and compute the average difference for each scenario.

Figure 4.3 and the Table in Appendix D show that most of the differences were negative, which means that $\hat{\lambda}$ was less than the simulated (real) λ. There are only a few combinations that have positive differences and they were all close to zero. Therefore, we can conclude that we are underestimating λ. Also, the results show that, for a given λ and D, the difference decreases as the sample size increases, and when D and n remain constant, the difference decreases as λ decreases. Figure 4.4 displays six plots of the number of components versus the difference with different sample sizes. It shows that in most scenarios, for a given λ and n, the difference

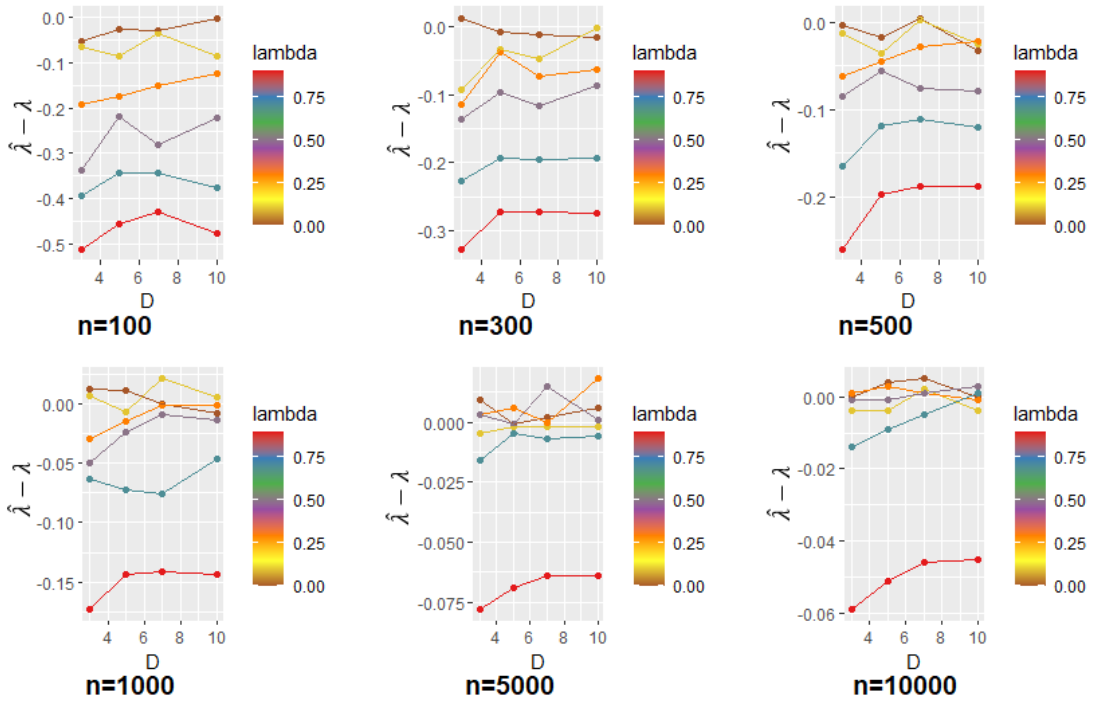

Figure 4.4: D vs $\hat{\lambda} - \lambda$ for different combinations of λ and n.

decreases as D increases.

4.3 Real-Life Data Analysis

In the following subsections we will compare cross-validation results for the LC, the ULC and Greenacre's transformation regression with three real-life datasets.

4.3.1 The Mortality dataset

"Mortality" is a 60×12 dataset. It consists of eight components of compositional predictor data, which are the proportions of eight various diseases in 60 countries. One variable is life expectancy as the response variable. The other three variables are no used in here. The eight diseases are infectious, neoplastic, endocrine, mental,

nervous, circulatory, respiratory, and digestive illnesses.

This dataset has no zeros in the compositional data. Therefore, we can use all three methods to analyze it. The mean square error ($MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$) of the three methods was 12.79 for Greenacre's transformation regression, 12.91 for the ULC, and 13.25 for the LC reression. There was no substantial difference among them based on MSE.

We then used cross-validation to compare which of the three has the lowest prediction error. Since the elements in each fold are chosen randomly, we ran the cross-validation 1000 times and computed the number of times that each method had the lowest cross-validation MSPE. The result was that LC regression had the lowest cross-validation MSPE 983 times, the ULC had the lowest cross-validation MSPE 15 times, and Greenacre's transformation regression had the lowest cross-validation MSPE 2 times. In summary LC regression method performs much better than the other two methods for this dataset even though it has the highest MSE of the three methods.

4.3.2 The lifeExpGdp dataset

The "lifeExpGdp" is a 27×9 dataset consisting of $D = 6$ part compositions, along with the life expectancy of men and women in 27 different countries in Europe (we only analyze the women's life expectancy). The compositional predictor data with $D = 6$ are the GDP of each country decomposed into six variables: agriculture, manufacture, construction, wholesale, transport and other. The dataset is accessible from the R package "*robCompositions*" [20].

This dataset has no zeros in the compositional data. Therefore, we will use all three methods to analyze it. The MSE of the three methods was 3.03 for Greenacre's transformation regression, 3.14 for the ULC, and 3.44 for the LC regression. Here the MSE of LC was somewhat larger than the other two, but there still was no substantial difference among them. Also, we find this result is the same as the previous real-life example; Greenacre's transformation regression has the lowest MSE, and LC has the highest MSE.

We then used cross-validation to compare the three with the lowest prediction error. The result was that LC regression had the lowest cross-validation MSPE 490 times, the ULC had the lowest cross-validation MSPE 226 times, and Greenacre's transformation regression had the lowest cross-validation MSPE 284 times. This result shows that the LC was still better than the other two methods, but ULC and Greenacre transformation regression were preferred about 25% of the times. In summary LC regression method performs much better than the other two methods. However, the two new methods had about a 25% chance of performing better than LC regression, which shows in some situations, the ULC and Greenacre's Transformation Regression may be better than LC regression.

4.3.3 The fgl dataset

The "fgl" is a 214×10 dataset consisting of $D = 8$ part compositions. The compositional predictor data are the weight percentages of oxides, which are Na (sodium), Mg (manganese), Al (aluminium), Si (silicon), K (potassium), Ca (calcium), Ba (barium) and Fe (iron). The response is the refractive index (refractive index determines how much the path of light is bent, or refracted, when entering a material). The data are also accessible from the R package "*MASS*" [21].

This dataset contained a lot of zeros (in total, 392 zeros, which is about 23% of the dataset) in the compositional data. Therefore, we cannot use the LC and ULC, and Greenacre's transformation regression is advantageous as it can accept zeros in the compositional data. We will use another method, namely α-PCR discussed in subsection 3.2.1, to compare with Greenacre's transformation regression. The MSE was 1.02 for Greenacre's transformation regression and 1.03 for the α-PCR. There was no substantial difference between them.

Next, we used cross-validation to compare which method has the lowest prediction error. We found that α-PCR regression had a lower cross-validation MSPE 985 times, and Greenacre's transformation regression had a lower cross-validation MSPE only 15 times. For this dataset, α-PCR performed much better than Greenacre's transformation regression.

Chapter 5

Conclusion and Future Work

This report reviewed existing regression models for compositional response or predictor data. We introduced two new regression methods, unconstrained log-contrast (ULC) regression and Greenacre's transformation regression for compositional predictor data. ULC regression is simpler than LC regression: no constraint of the coefficient $\boldsymbol{\beta}$. Greenacre's transformation regression has the advantage that it allows zeros in the compositional data.

Our simulation study results suggest that ULC, the unconstrained version of LC regression, can be used for an enormous sample size. LC generally has a lower prediction error when the sample size is small.

Greenacre's transformation regression's main advantage is that it can accept zeros in the compositional data. In a second simulation study examing how accurately λ can be estimated, we found that $\hat{\lambda}$ is biased using our method. Based on the real-life dataset, we found that this method may not be as accurate for prediction as the LC or ULC regression when there are no zeros in the compositional data. Also, not as precise as α-PCR when there are zeros in the compositional data. Further work

is needed to compare α-PCR and Greenacre's transformation regression through a simulation study. Also, another method of estimating λ could be explored, such as cross-validation to find an estimator of λ that is less biased.

Bibliography

[1] Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B(Methodological), 44(2), 139-177.*

[2] Aitchison, J. (1983). Principal component analysis of compositional data. *Biometrika, 70(1), 57-65.*

[3] Aitchison, J. (2003). The statistical analysis of compositional data. *New Jersey: Blackburn Press.*

[4] Egozcue, J.; Pawlowsky-Glahn, V.; Mateu-Figueras, G.; Barcelo-Vidal, C. (2003), Isometric logratio transformations for compositional data analysis, *Mathematical Geology, 35(3), 279–300.*

[5] Martìn-Fernández, J.; Barceló-Vidal, C.; Pawlowsky-Glahn, V. (2003). Dealing with zeros and missing values in compositional data sets. *Mathematical Geology, 35(3), 253-278.*

[6] Gueorguieva R.; Rosenheck, R. and Zelterman, D. (2008). Dirichlet component regression and its applications to psychiatric data. *Computational Statistics & Data Analysis, 52(12), 5344–5355.*

[7] Hijazi R. and Jernigan R. (2009). Modelling compositional data using Dirichlet regression models. *Journal of Applied Probability and Statistics, 4(1), 77–91.*

[8] Lancaster, H. (1965). The Helmert Matrices. *The American Mathematical Monthly, 72(1965), no.1, 4-12.*

[9] Hansen, B. E. (2019). Econometrics. *University of Wisconsin Department of Economics, pp:94-197.*

[10] Tsagris, M. (2015). Regression analysis with compositional data containing zero values. *Chilean Journal of Statistics, 6(2), 47-57.*

[11] Tsagris, M, and Stewart, C. (2018). A Dirichlet Regression Model for Compositional Data with Zeros. *Lobachevskii Journal of Mathematics, 39(3), 398-412.*

[12] Greenacre, M (2009). Power transformations in correspondence analysis. *Computational Statistics & Data Analysis, 53(8), 3107-3116.*

[13] Alenazi, A. (2021). Regression for compositional data with compositional data as predictor variables with or without zero values. *Journal of Data Science, 17(1), 219-237.*

[14] Hijazi,R. (2010). Dealing with rounded zeros in compositional data under Dirichlet models. *The Islamic Countries Society of Statistical Sciences, Lahore: Pakistan, 701-707.*

[15] Maclean's (2018). Canadian universities with the highest (and lowest) graduation rates. *https://www.macleans.ca/education/.*

[16] Pawlowsky-Glahn,V, Egozcue,J (2006). Compositional data and their analysis: an introduction. *Geological Society, London, Special Publications, pp:1-10*

[17] Bear,J. and Billheimer,D. (2016). A Logistic Normal Mixture Model Allowing Essential Zeros. *Austrian Journal of Statistics, 45(4), 3-23.*

[18] Tsagris, M., Preston, S. and Wood, A. (2011). A data-based power transformation for compositional data. *In Proceedings of the 4th Compositional Data Analysis Workshop, Girona, Spain.*

[19] Tsagris, M.; Athineou, G.;Alenazi, A.; Adam, C. (2022). Compositional: Compositional Data Analysis. R package version 5.6. https://CRAN.R-project.org/package=Compositional

[20] Templ,M. Hron, K. Filzmoser, P (2011). robCompositions: an R-package for robust statistical analysis of compositional data. In V. Pawlowsky-Glahn and A. Buccianti, editors, Compositional Data Analysis. Theory and Applications, pp. 341-355, John Wiley & Sons, Chichester (UK).

[21] Venables, W. & Ripley, B. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York.

[22] Weiss, N. A., & Weiss, C. A. (2016). Introductory statistics (10th ed.). *Pearson.*

Appendix A

Estimating the Power Transformation in Greenacre's Transformation Regression

```
bculcreg.mse.tune <- function(y, x, lambda.interval = c(-1, 0.9) ) {
fun <- function(lambda, y, x) {
X <- cbind(1, Compositional::green(x, lambda) )
be <- solve( crossprod(X), crossprod(X, y) )
mean( ( y - X %*% be )^2 )
}
runtime <- proc.time()
mod <- optimise(fun, lambda.interval, y = y, x = x)
runtime <- proc.time() - runtime
list(runtime = runtime, mspe = mod$objective,
best.lambda = mod$minimum)
}
```

Appendix B

Function for Simulation Study: LC vs ULC regression

```
function1<-function(n, D, disc, nfolds=10, R=1000){


##  disc is the distance from Ho ##.


Ftest<-matrix(rep(0,2*R),nrow=R)
Ftest<-as.data.frame(Ftest)
MSPEcom<-rep(3,R)
names(Ftest) <- c("F", "P-value")
for(i in 1:R) {


##bulit x##
x<-rdiri(n, c(rep(1/D,D)))
a<-matrix(runif(D*n, 1, 2),nrow=n)
x<-a/rowSums(a)
```

```
colnames(x)<-colnames(x, do.NULL = FALSE, prefix = "V")
logx<-log(x)

##random choose beta##
beta <- rep(0,D)
beta[1 : ceiling((D-1)/2)] <- runif((ceiling((D-1)/2)) ,1,2)
beta[(ceiling((D-1)/2)+1) : (D-1)]<-
runif((D-(ceiling((D-1)/2))-1) ,-2,-1)
beta[D]<- -sum(beta[1:(D-1)])
beta <- as.matrix(beta)+disc

##built y##
y1<-logx%*%beta
y<-y1+rnorm(n,mean=0,sd=sd(y1))

lc<-lc.reg(y,x)
ulc<-ulc.reg(y,x)
Ftest[i,]<-lcreg.aov(lc,ulc)

##MSPE compare##
lcmspe<-rep(0,nfolds)
ulcmspe<-rep(0,nfolds)
ina <- 1:n
folds <- Compositional::makefolds(ina, nfolds = nfolds,
stratified = FALSE, seed = FALSE)
nfolds<-length(folds)
for (j in 1:nfolds) {
```

```
xu <- as.matrix(x[folds[[j]],],ncol=D)
yu <- y[folds[[j]], ]
xa <- as.matrix(x[-folds[[j]],],ncol=D)
ya <- y[-folds[[j]], ]
lc1<-lc.reg(ya,xa,xnew=xu)
ulc1<-ulc.reg(ya,xa,xnew=xu)
lcmspe[j]<-mean((yu-lc1$est)^2)
ulcmspe[j]<-mean((yu-ulc1$est)^2)
}
if(mean(lcmspe)<mean(ulcmspe)) {
MSPEcom[i]<-1
} else if(mean(lcmspe)>mean(ulcmspe)) {
MSPEcom[i]<-0
} else {
MSPEcom[i]<-c("wrong")}
}

d<-sum(as.matrix(Ftest[,2])<0.05)
e<-sum(MSPEcom)
out <- list( Preject=d/R, Lcbetter=e/R)
return(out)
}
```

Appendix C

Function for Simulation Study: Greenacre's Transformation Regression

```
creatcoda<-function(n, D, lambda=c(0, 0.1, 0.3, 0.5, 0.7, 0.9)){
lambda<-as.data.frame(lambda)
result<-matrix(rep(0,dim(lambda)[1]*2),ncol=2)
for (i in 1:dim(lambda)[1]){

####x and z #####
a<-matrix(runif(D*n, 1, 2),nrow=n)
x<-a/rowSums(a)
z<-green(x,lambda[i])
colnames(x)<-colnames(x, do.NULL = FALSE, prefix = "V")

#### beta with first equal to 1 and others half are ####
####positive and half are negative  #####
```

```
beta <- rep(0,D)
beta[1] <- 1
beta[2 : ceiling(1+(D-1)/2)]<-
runif((ceiling(1+(D-1)/2)-1) ,1,2)
beta[(ceiling(1+(D-1)/2)+1) : D]<-
runif(D-(ceiling(1+(D-1)/2)) ,-2,-1)
beta <- as.matrix(beta)
#### built y #####
y1<-z%*%beta
y<-y1+rnorm(n,mean=0,sd=sd(y1))
#### Greenacre transformation regression #####
result[i,]<-c(bculcreg.mse.tune(y,x)$mspe,
bculcreg.mse.tune(y,x)$best.lambda)
}
out <- round(result, digit=2)
return(out)
}
##### run 1000 times to find the average ####
####  value of the best lambda. #####
resfirst<-matrix(rep(0,1000*6),nrow=6)

for (i in 1:1000){
resfirst[,i]<-creatcoda(100, 3)[,2]
}
n100d3<-cbind(rowMeans(resfirst),
matrix(c(0,0.1,0.3,0.5,0.7,0.9),nrow=6))
```

Appendix D

The Distance Between the simulated λ and optimal λ

No.	n	D	lambda	dis	No.	n	D	lambda	dis
1	100	3	0	-0.05	73	100	7	0	-0.03
2	100	3	0.1	-0.07	74	100	7	0.1	-0.04
3	100	3	0.3	-0.19	75	100	7	0.3	-0.15
4	100	3	0.5	-0.34	76	100	7	0.5	-0.28
5	100	3	0.7	-0.39	77	100	7	0.7	-0.34
6	100	3	0.9	-0.51	78	100	7	0.9	-0.43
7	300	3	0	0.01	79	300	7	0	-0.01
8	300	3	0.1	-0.09	80	300	7	0.1	-0.05
9	300	3	0.3	-0.12	81	300	7	0.3	-0.07
10	300	3	0.5	-0.14	82	300	7	0.5	-0.12
11	300	3	0.7	-0.23	83	300	7	0.7	-0.20
12	300	3	0.9	-0.33	84	300	7	0.9	-0.27
13	500	3	0	-0.00	85	500	7	0	0.00
14	500	3	0.1	-0.01	86	500	7	0.1	0.00

15	500	3	0.3	-0.06	87	500	7	0.3	-0.03
16	500	3	0.5	-0.08	88	500	7	0.5	-0.07
17	500	3	0.7	-0.16	89	500	7	0.7	-0.11
18	500	3	0.9	-0.26	90	500	7	0.9	-0.19
19	1000	3	0	0.01	91	1000	7	0	0.00
20	1000	3	0.1	0.01	92	1000	7	0.1	0.02
21	1000	3	0.3	-0.03	93	1000	7	0.3	-0.00
22	1000	3	0.5	-0.05	94	1000	7	0.5	-0.01
23	1000	3	0.7	-0.06	95	1000	7	0.7	-0.08
24	1000	3	0.9	-0.17	96	1000	7	0.9	-0.14
25	5000	3	0	0.01	97	5000	7	0	0.00
26	5000	3	0.1	-0.00	98	5000	7	0.1	-0.00
27	5000	3	0.3	0.00	99	5000	7	0.3	0.00
28	5000	3	0.5	0.00	100	5000	7	0.5	0.01
29	5000	3	0.7	-0.02	101	5000	7	0.7	-0.01
30	5000	3	0.9	-0.08	102	5000	7	0.9	-0.06
31	10000	3	0	0.00	103	10000	7	0	0.00
32	10000	3	0.1	-0.00	104	10000	7	0.1	0.00
33	10000	3	0.3	0.00	105	10000	7	0.3	0.00
34	10000	3	0.5	-0.00	106	10000	7	0.5	0.00
35	10000	3	0.7	-0.01	107	10000	7	0.7	-0.00
36	10000	3	0.9	-0.06	108	10000	7	0.9	-0.05
37	100	5	0	-0.03	109	100	10	0	-0.00
38	100	5	0.1	-0.09	110	100	10	0.1	-0.09
39	100	5	0.3	-0.17	111	100	10	0.3	-0.12
40	100	5	0.5	-0.22	112	100	10	0.5	-0.22
41	100	5	0.7	-0.34	113	100	10	0.7	-0.38

42	100	5	0.9	-0.46	114	100	10	0.9	-0.48
43	300	5	0	-0.01	115	300	10	0	-0.02
44	300	5	0.1	-0.03	116	300	10	0.1	-0.00
45	300	5	0.3	-0.04	117	300	10	0.3	-0.06
46	300	5	0.5	-0.10	118	300	10	0.5	-0.09
47	300	5	0.7	-0.19	119	300	10	0.7	-0.19
48	300	5	0.9	-0.27	120	300	10	0.9	-0.28
49	500	5	0	-0.02	121	500	10	0	-0.03
50	500	5	0.1	-0.04	122	500	10	0.1	-0.02
51	500	5	0.3	-0.04	123	500	10	0.3	-0.02
52	500	5	0.5	-0.06	124	500	10	0.5	-0.08
53	500	5	0.7	-0.12	125	500	10	0.7	-0.12
54	500	5	0.9	-0.20	126	500	10	0.9	-0.19
55	1000	5	0	0.01	127	1000	10	0	-0.01
56	1000	5	0.1	-0.01	128	1000	10	0.1	0.01
57	1000	5	0.3	-0.01	129	1000	10	0.3	-0.00
58	1000	5	0.5	-0.02	130	1000	10	0.5	-0.01
59	1000	5	0.7	-0.07	131	1000	10	0.7	-0.05
60	1000	5	0.9	-0.14	132	1000	10	0.9	-0.14
61	5000	5	0	-0.00	133	5000	10	0	0.01
62	5000	5	0.1	-0.00	134	5000	10	0.1	-0.00
63	5000	5	0.3	0.01	135	5000	10	0.3	0.02
64	5000	5	0.5	-0.00	136	5000	10	0.5	0.00
65	5000	5	0.7	-0.00	137	5000	10	0.7	-0.01
66	5000	5	0.9	-0.07	138	5000	10	0.9	-0.06
67	10000	5	0	0.00	139	10000	10	0	0.00
68	10000	5	0.1	-0.00	140	10000	10	0.1	-0.00

69	10000	5	0.3	0.00	141	10000	10	0.3	-0.00
70	10000	5	0.5	-0.00	142	10000	10	0.5	0.00
71	10000	5	0.7	-0.01	143	10000	10	0.7	0.00
72	10000	5	0.9	-0.05	144	10000	10	0.9	-0.04

Vita

PERSONAL DETAILS

Name: Zhenduo Huang

EDUCATION

2005-2010 Bachelor degree of Civil Engineering. UNB
2015-2017 Master degree of Mathematics. UNB