University of California

Los Angeles

Logistic Gifi: A Logistic Distance Association Model for Exploratory Analysis of Categorical Data

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Gary William Evans

2014

Abstract of the Dissertation

Logistic Gifi: A Logistic Distance Association Model for Exploratory Analysis of Categorical Data

by

Gary William Evans

Doctor of Philosophy in Statistics

University of California, Los Angeles, 2014

Professor Jan de Leeuw, Chair

In this work, we explore a distance association method, Logistic Gifi, for categorical data which advances geometric data analysis techniques in much the same way that homogeneity analysis did with regard to correspondence analysis, multidimensional scaling, and general clustering methods. It uses the methods of multidimensional unfolding with a probability-based loss measure to create low-dimensional geometric representations of data in which distances correspond in a direct way to the probabilistic structure of the data. As with homogeneity analysis, a central feature of our method is the use of binary indicator matrices and, in some applications, fuzzy-coded (i.e., non-binary, row stochastic) indicator matrices to represent categorical data. This gives us a very versatile method with considerable flexibility in the types of data which can be analyzed. We create and study algorithms to use the method to compute low-dimensional geometric representations of various types of data. We analyze the convergence properties of this complex algorithm and show how minimal polynomial extrapolation can be used to accelerate it. We then study relationships between this logistic distance method and logit-based regressions. We present several applications of the method to visualizations of regression results as well as data types such as roll calls, social networks, and Markov chains. Finally, a version of the method with bias parameters is introduced and developed and used to emphasize features of data visualizations. We show how bias constraints can be used to represent certain types of model testing. Noting the similarities between the model with bias parameters and ideal point discriminant analysis, we examine these using bias constraints and different forms of indicator matrices. Last, we study the stability of the method.

The dissertation of Gary William Evans is approved.

Jeffrey B. Lewis

Frederick Paik Schoenberg

Mark Handcock

Jan de Leeuw, Committee Chair

University of California, Los Angeles

2014

To my wife, Yuan, and son, Colin . . .

as much as this work takes

of my mind, there, you are first

and uppermost always

Table of Contents

List of Figures

List of Tables

Acknowledgments

Considering where this journey began for me,
adept, expert guides were essential. My sincerest thanks go to
Professors Wu, Xu, Hansen, Schoenberg, Ferguson, Zhou, Yuille,
Bentler, Lange, Gould, Cristou, and Handcock who were every bit this and more.
And, of course, my special thanks to Professor Jan de Leeuw whose work
and guidance have been unceasing sources of inspiration to me.

Vita

1985 B.S. (Mathematics) and B.S. (Secondary Education), Penn State University, University Park, CA.

1988 M.A. (Mathematics), Indiana University, Bloomington, IN

1991 J.D. Villanova University, Villanova, PA

2009–present Teaching Fellow, Statistics Department, UCLA. (Discussion and lab assistant for sections of Statistics 10 (Introduction to Statistics) and Statistics 112 (Statistics for the Social Sciences) and Instructor (Winter, 2013) for Statistics 13 (Statistics for the Life Sciences) and (Summer, 2013) Statistics 100A (Introduction to Probability - Upper Division) under direction of Professor Rob Gould.)
Most Outstanding Computational Statistician, 2011-2012

Publications

Exploratory Multivariate Analysis by Example Using R, Journal of Statistical Software, Vol. 40, Book Review 2, April, 2011.

CHAPTER 1

Introduction

We will first look at some basic ideas of distance association methods using indicator matrices. Then we will introduce what we have referred to as a logistic distance association model, Logistic Gifi, and discuss some of its basic properties. We will close this section by looking at two basic demonstrations of the model; the first using binary indicator data, the second using fuzzy-coded indicator data.

We begin with a simple example to illustrate the construction of indicator matrices and the aim of the method. Suppose we have n persons each asked to select their one favorite among m categories - of music, for example (with, generally, m much smaller than n). The results of this survey can be represented in an $n \times m$ indicator matrix, G, with each row containing a single 1 and m-1 zeroes. Also, we assume that each type of music is chosen at least once; i.e, each column of the matrix contains at least one 1 (otherwise, we can simply discard that category from the data). Obviously, each person is identified with 1 category, the group of persons is partitioned by category membership, and, conversely, each category can be identified with a group of 1 or more persons.

Next, suppose that we have some other partitionings of the persons based on some other extrinsic characteristics, such as gender or age, for instance. These can, likewise, be represented by indicator matrices. It is not likely that each of these will partition the group of persons identically to the music preference matrix. So now, each person has a profile based on their category memberships and each music category has a somewhat more complex profile based on the persons choosing that type of music and some weighting of their other category memberships. Thus, we become interested in modeling the probabilities that persons with particular profiles will prefer music of a certain type and we want a geometric representation to reflect this aspect of the data. Further, we hope to do so in such a way that the probabilities can be derived directly from the distances between points in the geometric configuration.

This basic example motivates the distance model we shall examine, named "Logistic Gifi" (LG) by its inventor, De Leeuw (2005) [15]. The approach advances geometric data analysis techniques in much the same way that De Leeuw's homogeneity analysis did with regard to correspondence analysis (CA), multidimensional scaling (MDS), and general clustering methods. (A detailed treatment of this can be found in Gifi (1990) [33], with an excellent summary in Michaelidis & De Leeuw (1998) [49].)

We will first consider a version of LG using Euclidean distance (d(x, y)) as the plotting metric. For a single n x m binary indicator matrix, G, as we have postulated above, which models the data as the realization of n multinomial trials, we let:

$$(1) \qquad \pi_{il}(\mathrm{X,Y}) = \frac{exp(-d(x_i,y_l))}{\sum_{j=1}^{m} exp(-d(x_i,y_j)))}$$

where π_{il} is the probability that object i is in (or has chosen) category l, x$_i$ and y$_j$ are, respectively, the ith and jth rows of the coordinate matrices X and Y; that is, X and Y are respectively, $n \times d$ and $m \times d$ matrices with each row of X giving the coordinates of a person (which we shall refer to from here on as an object) in the d-dimensional geometric representation of the data we seek and each row of Y the coordinates of a category.[1] We refer to this as Unbiased[2] Euclidean LG, denoted uLG-1, to distinguish from the general LG model which includes weighting, or bias, parameters, and, as noted, may involve difference distance metrics.[3]

Notice the resemblance of the uLG-1 modeling function to the logit link function used in logistic regression. Hence, the name given by De Leeuw to the model.[4] The general model further betrays an affinity with the the ideal point discriminant analysis (IPDA) of Takane et al. (1987) [63], and the log-linear modeling of contingency table data, of which a thorough review is presented in De Rooij (2001) [25]. (Each of these topics will be discussed in some detail in later sections; IPDA in Chapter 5 on LG with Bias Parameters and log-linear modeling in Chapter 2 on Categorical Longitudinal Data and Model Testing and in Chapter 5.) As will be seen, with LG, De Leeuw has connected these techniques with the more

[1] Though sometimes higher dimensional models are useful to consider, unless otherwise required or specified, we will choose d = 2 for two main reasons. First, this provides ease of display and interpretation and, second, in most scaling models, when a 1-dimensional representation is not viable, 2 dimensions, occasionally 3, have usually been found to give adequate model fit. See Heiser & Meulman, 1986 [40].

[2] We use the term Unbiased here and throughout to mean without bias parameters, not to refer to expectations of parameter estimates.

[3] In the most general form of LG, which we refer to as Biased LG (to mean LG with bias parameters), for a single $n \times m$ indicator matrix, G, we model the probability that object i is in category l as:

$$\pi_{il}(\mathrm{X,Y}) = \frac{\beta_l exp(-\phi(x_i,y_l))}{\sum_{j=1}^{m} \beta_j exp(-\phi(x_i,y_j))}$$

where β's are the bias parameters and $\phi(\mathrm{x}_i,\ \mathrm{y}_l)$ is a function giving a distance between object i and category j. In the uGL-1, all β's = 1 and ϕ is Euclidean distance. We will discuss the model with bias parameters and some of its uses in sections to follow.

[4] In addition to Euclidean distance, DeLeeuw proposes as well the use of squared Euclidean distances and inner products in the LG model. Squared-distances were used in the original development of MDS since their derivatives are much easier to compute and work with. In log-linear modeling, they are typically used for the same reason, since such modeling is generally done by maximum-likelihood methods requiring the computation of an information matrix. With the development of the majorization method we will work with, known as SMACOF, it is no longer necessary to work with squared distances. Since, in visualization applications, Euclidean distances are easier to directly assess and interpretation of distances is essentially the same, we will study only Euclidean distance models in this work. As for the inner product model, we have found that interpretation of these types of configurations is often quite difficult. For this reason, again, in this work we will forego their study in favor of Euclidean distance. Importantly, as De Leeuw (2006) [17] notes, by suitably adjusting the bias parameters, fitting a squared distance model with bias is equivalent to fitting an inner product model. A probit-linked distance model is also proposed by De Leeuw

general distance methods of MDS and, in particular, multidimensional unfolding (MDU). Through the use of indicator matrix data considered as (or, in the case of fuzzy indicators, given as) measures of probabilities, this allows for the techniques to be examined in a broader range of data analysis contexts. Also, within the framework of MDU, we can view LG as having been developed to address some of De Leeuw's concerns about this method.

With the model as set out above, the goal is to find coordinates (x_{i1}, x_{i2}) for each object i = 1, 2,...,n and (y_{j1}, y_{j2}) for each category j = 1, 2,...,m, so that the probabilities computed by (1) using the resulting distances between points are as close to the data as possible. Since we are using the negative of distances for powers, the optimum coordinates will be such that objects are closest to the categories they belong to or select and relatively farther from the other categories. A key feature of the design is that persons and categories with similar profiles will be located close together. Some preliminary observations are helpful in further understanding uLG-1.

First, for data which categorizes the objects on a single variable (i.e., data represented by a single binary indicator matrix), to have a perfectly fitting model we must have $\mathrm{d}(\mathrm{x}_i,\mathrm{y}_j) = 0$ for all i, j such that object i is in category j (i.e., for $g_{ij} = 1$) and $\mathrm{d}(\mathrm{x}_i,\mathrm{y}_j) = \infty$ otherwise. Notice that for a single variable with two or three categories, this is theoretically possible in 1 dimension and in 2 dimensions several categories can be accommodated. We are, however, interested in realizable, as opposed to idealized, geometric representations of data, so we will not accept plots with infinite distances. Nonetheless, even with this restriction, it is easy to see that for any finite number of objects measured on any single variable represented by a binary indicator matrix (with a finite number of categories), a 1-dimensional solution can be found to any desired degree of point-wise precision.[5]

[5] It is also easy to see, for example, that this is true for classifications involving more than 1 variable if they each have the same number of categories and these identically partition the objects. From the point of view of LG, such variables are essentially identical.

An important result of De Leeuw (2006) [14] extending this observation is that, for multiple binary variables, each object can be plotted closest to all categories to which it belongs, if and only if, deviance can be made arbitrarily close to 0. That is to say, the minimization will find coordinates so that the model probabilities will approximate the data to any desired uniform point-wise precision. We refer to this as the Ideal Model Theorem (IMT).

We give a proof of the IMT here for Unbiased LG with Euclidean distance. The brief discussion above on the basics of LG gives an indication of how to proceed. First, suppose we have a configuration in which each object is closest the category to which it belongs for all variables. Without loss of generality, to simplify the notation somewhat, we consider one variable at a time. We have:

(2) $\pi_{il}(\mathrm{X,Y}) = \frac{exp(-d(x_i,y_l))}{\sum_{j=1}^{m} exp(-d(x_i,y_j))}$,

and we want to show that we can obtain a configuration such that (2) can be made arbitrarily close to 1 for all i and l such that object i is in category l. To further simplify the notation, assume object i is in category 1 (since some object must be). Notice that, by the assumed distance condition, we have, for j $\geq$ 2:

(3) $\mathrm{d}(\mathrm{x}_i,\ \mathrm{y}_j) = \mathrm{d}(\mathrm{x}_i,\ \mathrm{y}_1) + \mathrm{c}_j$, where all $\mathrm{c}_j > 0$.

Thus, $\exp(-\mathrm{d}(\mathrm{x}_i,\ \mathrm{y}_j)) = \exp(-\mathrm{d}(\mathrm{x}_i,\ \mathrm{y}_1) - \mathrm{c}_j) = \exp(-\mathrm{d}(\mathrm{x}_i,\ \mathrm{y}_1))\exp(-\ \mathrm{c}_j)$ and, by factoring out $\exp(-\mathrm{d}(\mathrm{x}_i,\ \mathrm{y}_1))$, (2) becomes

(4) $\pi_{i1} = \frac{1}{1+\sum_{j=2}^{m} exp(-c_j)}$.

Let $\epsilon > 0$ be arbitrarily small. To have 1 - $\pi_{i1} < \epsilon$, it is clear that we must produce a configuration satisfying the distance condition for which the c$_j$'s are increased so that $\sum_{j=2}^{m} exp(-c_j)$ is sufficiently close to 0. We can do this by a dilation of the configuration; i.e., by multiplying all coordinates of the given configuration by some constant D >1. It is easily seen that doing this has the effect of multiplying all object-to-category distances by the same factor, D, and it preserves the distance condition. So, we must find D so that:

$$(5) \qquad 1 - \frac{1}{1+\sum_{j=2}^{m} exp(-Dc_j)} < \epsilon;$$

or, after a little algebra:

$$(6) \qquad \sum_{j=2}^{m} exp(-Dc_j) < \frac{\epsilon}{1-\epsilon}.$$

Since $\lim_{x \to +\infty} e^{-x} = 0$, and ϵ is fixed, the dilation constant D can be chosen so that all *m-1* terms of the sum on the left of (6) are less than $(\frac{\epsilon}{1-\epsilon})(\frac{1}{m-1})$. Thus, (6) is true. Now, since the object and category are randomly chosen both from a finite number of possibilities and since ϵ is arbitrarily small and D can be arbitrarily large, it follows that D can be chosen so that the APWL over all variables can made as small as possible. The *only if* direction is obvious, particularly from the preceding demonstration, and this gives the result. Notice that it is clearly seen that this proof applies to squared Euclidean distances and inner product distances since the effect of dilation is the same.[6]

These properties of LG we have been discussing are mainly due to two facts: first, as noted above, objects should be closest to the categories they are in, but,

[6] An interesting implication of the IMT is that if another method, such as e.g. `homals`, produces a 100% classification configuration, that configuration can be dilated to create an LG plot with arbitrarily small APWL. Thus, with very little extra work, the data can be simultaneously studied from the point of view of both methods. With `homals`, for example, the notions of discrimination measure and correlation of variables which are features of that method can be analyzed in connection with LG.

for binary indicators, they do not need to be identically far from the other categories; second, the distances from object to object and from category to category are not directly involved in (1). Thus, to increase the precision of the model, certain objects and categories (those with relatively unique profiles) can be moved large distances from the main groups in the configuration. It is hard to overemphasize that only distances from objects to categories are used in the probability computations. An important implication of this is that, as noted above, the method somewhat resembles MDU; thus, MDU theory plays a significant role in the development and study of LG.

Considering the above observations, we further see that, in general, the coordinates giving us desired precisions, or even exact solutions where those are available, are not unique. They can be changed by rotation, translation, or reflection and the overall configurations of two equally well-fitting plots can, in theory, be quite different. In examining an LG plot, the focus, then, is generally to be on the distance differences between object and category points, the clustering of between and within object and category groups, and on basic geometric patterns in the final configuration. Interpretation is based heavily on object and category profiles, such as in CA; however, interpretation of the dimensions or principal axes of the plot, though possibly informative, is secondary.

Turning back to our music selection example, suppose now that we have binary indicator matrices which classify these objects (persons), perhaps, as mentioned previously, by age, gender, and ethnicity, etc.[7] It is not now always possible to find 1-dimensional coordinates for a given point-wise precision and may not be possible to find 2-dimensional ones. We need a procedure, therefore, which will find coordinates to minimize the overall discrepancies between the data and

[7] Age can be considered a categorical designation, of course, by classifying persons as being in a particular age interval. In this way, such a classification, and continuous variables in general, can be modeled, or *coded*, by binary indicator matrices. Such matrices can be used to code various other types of data, as well. Chapter 2 of Gifi (1990) [33] has a thorough discussion of this topic and also presents three methods for coding missing data.

model probabilities. We can think of this, in the parlance of MDS, as the stress of the model. However, as noted above, De Leeuw's work in MDS and MDU led him to be skeptical of the traditional stress functions. He sought an approach which would not rely on arbitrary normalizations to avoid degenerate solutions and the results of which could be assessed based on a more reliable and readily interpretable measure than traditional stress. See De Leeuw (2007) [19] and De Leeuw & Heiser (1982) [22] for interesting discussions of these points.

The approach taken by De Leeuw [15] is to view the data as arising from a multinomial or product multinomial distribution and to minimize the negative log-likelihood, or deviance, of the model. As De Leeuw notes, this is not necessarily a realistic description of the data generating mechanisms in such studies, but it is a useful conceptualization. It allows for fairly direct, likelihood-based computations and interpretations since distances model probabilities, instead of dissimilarities or abstract preferences, as they do in classic MDS and MDU. Of particular interest is that the algorithmic computation of the configuration uses MDU, leading to applications of MDU in contexts outside of the realm of psychometrics.

1.1 The LG Algorithm

Let us turn now to that computation. Consider the most general case, where we have k variables with variable j = 1,2,...,k having m$_j$ categories and each variable measured on n objects. We can form what we call a *super-indicator* matrix, G, by column-binding the indicator matrices for each of the k variables. We must find coordinates now for each of the $\sum_{j=1}^{k} m_j$ categories and each of the n objects. Thus, we have to set initial positions for, and ultimately determine solutions for, k category coordinate matrices, Y$_j$ and for the objects (call their coordinate matrix, X). For notational convenience, we row-bind the category coordinate matrices into a single coordinate matrix, Y. With our data and parameters in this form, we use

De Leeuw's multinomial distribution concept to write the likelihood (assuming independence) as:

(7) $\prod_i^n \prod_j^k \prod_l^{mj} \pi_{ijl}(\mathrm{X,Y})^{g_{ijl}}$

Thus, the negative log-likelihood (or deviance) is:

(8) $\Delta(\mathrm{X,Y}) = \text{-}\sum_i^n \sum_j^k \sum_l^{mj} g_{ijl} \log \pi_{ijl}(\mathrm{X,Y})$

To find configuration coordinates that will minimize this function, we use quadratic majorization[8] to reduce the problem to an iterated least squares problem. We will show that in each iteration we must minimize:

(9) $\sum_i^n \sum_j^k \sum_l^{mj} (\mathrm{d}(\mathrm{x}_i, \, \mathrm{y}_{jl}) \text{ - } \tilde{z}_{ijl})^2$

where

(10) $\tilde{z}_{ijl} = \mathrm{d}(\tilde{x}_i, \, \tilde{y}_{jl}) \text{ - } 4(g_{ijl} \text{ - } \pi_{ijl}(\tilde{X}, \, \tilde{Y}))$

$\tilde{X}$ and $\tilde{Y}$ being coordinate matrices from the prior iteration of the algorithm and the target, $\tilde{z}$, changing with each iteration. We follow the derivation of this result sketched by De Leeuw (2005b) [16].

We begin with a basic description of the majorization-minorization (MM) optimization method.

Suppose we wish to iteratively minimize a complicated function f:$\mathbb{R}^m \rightarrow \mathbb{R}$. A

[8] For excellent discussions of majorization and, in particular, quadratic majorization, see Lange (2004) [45] and De Leeuw & Lange (2009) [23]. In the contexts of MDS and MDU, see also De Leeuw & Mair (2009) [24]. This latter work is essential to what is to follow here.

function g(x | x_m) is said to *majorize* f at the support point x_m if:

(11) $f(x_m) = g(x_m \mid x_m)$, and

(12) $f(x) \leq g(x \mid x_m)$ for all $x \neq x_m$.

Thus, g is tangent to f at x_m and above it elsewhere in their common domain.

It can be shown that, if x_{m+1} is the minimum of g(x | x_m), then

(13) $f(x_{m+1}) \leq f(x_m)$.

So, if a simple majorizing function g(x | x_{m+1}) can be found in a closed form dependent in some way on the function f, we can minimize f by successively minimizing the majorizing functions. That is, we will have an iterative algorithm which finds a sequence of points {x_m} in the domain of f such that (13) holds at every iteration. If f is bounded below (as when, for example, f:$\mathbb{R}^m \rightarrow (0, \infty)$), the sequence of points {f(x_m)} converges to a minimum. This descent property gives the MM method a high degree of numerical stability.

It is easy to show that majorization is closed under summation; i.e., if g_1 majorizes f_1 and g_2 majorizes f_2, then $g_1 + g_2$ majorizes $f_1 + f_2$. This property is useful for majorizing log-likelihoods, as we shall see.

Finding a suitable majorizing function is not always easy. However, an important result (which we will refer to here as the quadratic majorization theorem) states that, for f:$\mathbb{R}\rightarrow\mathbb{R}$, if f is twice differentiable and there is a B >0 such that f"(x) $\leq$ B for all x, then for each y the convex quadratic function

(14) $$\text{g(x) = f(y) + f'(y)(x-y) + } \tfrac{1}{2}\text{B(x-y)}^2$$

majorizes f at y. This is easily proved by considering the Taylor series expansion of f at y. See De Leeuw & Lange (2009) [23]. A more general version of this theorem for functions on $\mathbb{R}^m$ is that

(15)) $$\text{g(x) = f(y) + } d\text{f(y)(x-y) + } \tfrac{1}{2}\text{(x-y)}^t\text{B(x-y)}$$

majorizes f(x) at y, where B is a positive definite matrix such that B - d^2f(x) is positive semi-definite. See Lange (2004) [45] and Böhning & Lindsay (1988) [5] We can apply this theorem and the closure over summation property to the terms of the LG link function (1) to obtain a majorizing function.

Now, for the LG link function, considered for 1 categorical variable we have from (1),

$$\pi_{il}(\text{X,Y}) = \frac{exp(-\phi(x_i,y_l))}{\sum_{j=1}^{m} exp(-\phi(x_i,y_j))}.$$

Thus:

(16) $$\frac{\partial\pi_{il}}{\partial\phi_{il}} = \pi_{il}^2 - \pi_{il},$$

and,

(17) $$\frac{\partial\pi_{il}}{\partial\phi_{iv}} = \pi_{il}\pi_{iv}.$$

Here, ϕ represents whatever distance function we choose for the LG link. In our applications, it is the Euclidean distance function, but notice as we proceed that, for this derivation, it need not be. It could represent squared distances or even some non-Euclidean metric. It need not even satisfy the definition of a distance function. So, in general, we have:

$$\frac{\partial \pi_{il}}{\partial \phi_{iv}} = \pi_{il}\pi_{iv} - \pi_{il}\delta^{lv} \tag{18}$$

where δ^{lv} is the Kronecker delta equal to 1 if $l = v$ and 0 otherwise.

From this, it follows that for the negative log-likelihood (8), we have:

$$\frac{\partial \Delta}{\partial \phi_{il}} = g_{il} - \pi_{il} \tag{19}$$

$$\frac{\partial^2 \Delta}{\partial \phi_{il} \partial \phi_{iv}} = \pi_{il}\delta^{lv} - \pi_{il}\pi_{iv}. \tag{20}$$

Thus, $d^2\Delta$ is a matrix of the form $H = \Pi - \pi\pi^t$ where π is some probability vector and Π is the diagonal matrix with the vector π on the diagonal. The largest eigenvalue λ_1 of this matrix being bounded above by any matrix norm (see eg. Gentle (2007) [32]), we have:

$$\lambda_1 \leq \max_{i=1}^{n} \sum_{j=1}^{n} |h_{ij}| = \max_{i=1}^{n} 2\pi_i(1 - \pi_i) \leq \frac{1}{2} \tag{21}$$

i.e.; the sums of the row absolute values are of the form $2\pi_i(1 - \pi_i)$ and, thus, bounded by $\frac{1}{2}$. Thus, the positive definite matrix B = $\frac{1}{2}I$ is such that B - $d^2\Delta$ has all real, non-negative eigenvalues and is, therefore, positive semi-definite.

Notice we have examined this for a single variable, but additional variables simply add summands of the same form to the partials of Δ. Applying the general quadratic majorization theorem stated above and the closure property of majorizations for sums to this multivariate form of Δ (expanding around the distance matrices $\phi_{ijl}(\tilde{X},\tilde{Y})$) and grouping terms, we have:

$$(22) \qquad \Delta(\mathrm{X,Y}) \leq \Delta(\tilde{X},\tilde{Y}) + \textstyle\sum_i^n \sum_j^k \sum_l^{mj} (g_{ijl} \text{ - } \pi_{ijl}(\tilde{X},\tilde{Y}))(\phi_{ijl}(\mathrm{X,Y}) \text{ - } \phi_{ijl}(\tilde{X},\tilde{Y}))$$
$$+ \tfrac{1}{4}\textstyle\sum_i^n \sum_j^k \sum_l^{mj} (\phi_{ijl}(\mathrm{X,Y}) \text{ - } \phi_{ijl}(\tilde{X},\tilde{Y}))^2$$

where $\tilde{X}$ and $\tilde{Y}$ are given coordinate matrices at a particular iteration of the majorization. The next step is to view the right side of (22) as a sum of several quadratic expressions and to complete the square of each one. This is done by dividing both sides of (22) by $\frac{1}{4}$, then adding $4((g_{ijl} \text{ - } \pi_{ijl}(\tilde{X},\tilde{Y}))^2$ to each quadratic term. Notice that neither the division by a positive number nor the adding of a square (hence, non-negative) term disturbs the majorization inequality. Regrouping terms and dividing both sides back by 4 again, gives:

$$(23)\ \Delta(\mathrm{X,Y}) \leq \Delta(\tilde{X},\tilde{Y}) + \tfrac{1}{4}\textstyle\sum_i^n \sum_j^k \sum_l^{mj} ((\phi_{ijl}(\mathrm{X,Y}) \text{ - } \phi_{ijl}(\tilde{X},\tilde{Y})) + 2(g_{ijl} \text{ - } \pi_{ijl}(\tilde{X},\tilde{Y})))^2$$
$$\text{- } 2 \textstyle\sum_i^n \sum_j^k \sum_l^{mj} (g_{ijl} \text{ - } \pi_{ijl}(\tilde{X},\tilde{Y}))^2.$$

Now it is evident that minimizing (23) amounts to minimizing the middle term on the right of the inequality since, at any given iteration, the other terms are constants. Rewriting the middle term as

$$(24) \qquad (\phi_{ijl}(\mathrm{X,Y}) \text{ - } \phi_{ijl}(\tilde{X},\tilde{Y}) + 2(g_{ijl} \text{ - } \pi_{ijl}(\tilde{X},\tilde{Y})))^2$$

it is clear that the minimum occurs at the coordinates X and Y which give:

$$\phi_{ijl}(\mathrm{X,Y}) = \tilde{z}_{ijl} \tag{25}$$

where

$$\tilde{z}_{ijl} = (\phi_{ijl}(\tilde{X},\tilde{Y}) \text{ - } 2(g_{ijl} \text{ - } \pi_{ijl}(\tilde{X},\tilde{Y})). \tag{26}$$

Notice that the majorization can be applied to LG with Bias Parameters as well simply by writing ϕ(X,Y) = d(X,Y) + log(β) where d is the chosen distance function. This does not change the derivation above.

Finally, since the diagonal elements of H are bounded above by $\frac{1}{4}$ and the off-diagonal elements by 0, we can use local bounds, combined with generalized block-relaxation, to bound H by $\frac{1}{4}I$. See De Leeuw (1994) [21]. Using the same calculations as above, we get the majorization:

$$\begin{aligned}\Delta(\mathrm{X,Y}) \leq \Delta(\tilde{X},\tilde{Y}) + \textstyle\sum_i^n\sum_j^k\sum_l^{mj} (g_{ijl} \text{ - } \pi_{ijl}(\tilde{X},\tilde{Y}))(\phi_{ijl}(\mathrm{X,Y}) \text{ - } \phi_{ijl}(\tilde{X},\tilde{Y}))\\ + \tfrac{1}{8}\textstyle\sum_i^n\sum_j^k\sum_l^{mj}(\phi_{ijl}(\mathrm{X,Y}) \text{ - } \phi_{ijl}(\tilde{X},\tilde{Y}))^2\end{aligned} \tag{27}$$

which yields the target minimum:

$$\tilde{z}_{ijl} = (\phi_{ijl}(\tilde{X},\tilde{Y}) \text{ - } 4(g_{ijl} \text{ - } \pi_{ijl}(\tilde{X},\tilde{Y})). \tag{28}$$

This gives a slightly sharper majorization and slightly improved performance from our algorithm.

A little parsing of these equations gives rise to a straightforward approach to constructing an algorithm for computing coordinates in uLG-1. Suppose we have some proposed configuration for objects and categories; either a starting configuration or one produced as an iterate of our algorithm. From the coordinates of this configuration, we can compute, for each variable, the object-to-category distance matrix, then column-bind these into a super distance matrix, d($\tilde{X}$,$\tilde{Y}$). From (26) and (28), it is obvious that deviance is decreased and ultimately minimized by a configuration with super distance matrix equal to the target, -$\tilde{z}_{ijl}$. This new distance matrix is obtained by pointwise subtraction of four times the difference between indicator and model probabilities from d($\tilde{X}$,$\tilde{Y}$).

This seems quite simple, but note that we are not really seeking a distance matrix, even one with perfect or near-perfect model fit. We point out here that these can fairly easily be obtained by observing the following: Let p$_{ij}$ be the *i,j*th entry in the indicator matrix; i.e., p$_{ij}$ is the probability that subject *i* chooses or is in category *j*. Now, suppose that from the data we have an estimate of the distances between the objects and category 1 - call them d$_{i1}$'s - that satisfies the basic ordinal condition that d$_{i1}$ $<$ d$_{j1}$ if and only if p$_{i1}$ $>$ p$_{j1}$ (i.e., large probabilities correspond to small distances and vice versa) and, perhaps, some basic metric condition, as well. From these distances, we can easily compute the distances between the objects and all other categories that must exist in an ideal (i.e., perfect fitting) LG model by the simple formula:

$$(29) \qquad \mathrm{d}_{ik} = \mathrm{d}_{i1} + \log(\mathrm{p}_{i1}) \text{ - } \log(\mathrm{p}_{ik}).$$

For fuzzy indicator matrices with all non-zero entries, the terms of this equation are all well-defined. For binary matrices, we need to replace zero probabilities with some small $\varepsilon > 0$. Though well-defined arithmetically, note that d$_{ik}$ may then be negative, which we do not consider to be feasible distances. To cure this problem, notice that to each distance, we may, without changing the model probabilities given by (1), add a positive constant (the absolute value of the minimum negative distance, for example) to make all distances in the matrix non-negative. We now have a distance matrix that will exactly model the indicator probabilities (for positive fuzzy indicators) or, for binary indicators, as near-exactly as we wish, depending on our choice of ε.

What we are seeking, however, is a configuration; i.e., a set of coordinates, which produces such a distance matrix between them. The method for doing this from a full distance matrix is MDS; from a partial distance matrix as we have in LG, it is MDU. For a full discussion of these techniques, see Borg & Groenen (2005) [6]. Therefore, the next step in the LG algorithm must be to use MDU to compute object and category coordinates that give the object-to-category distances found in the -$\tilde{z}_{ijl}$ matrix. We will then iterate this process until we have reached our desired precision. This we measure by average pointwise loss (APWL) since pointwise loss is directly involved in the algorithm and since it is more easily interpretable in this context than likelihood or log-likelihood.

Here, we have to be prepared to deal with 2 issues. First, the -$\tilde{z}_{ijl}$ matrix may contain negative entries. There are existing MDU algorithms we can employ to compute coordinates, but these are not well-behaved when the distances to be fitted are negative. Fortunately, we can deal with this, as discussed above, by adding a constant to the distance matrix for each variable to make all distances non-negative. Second, the distances, even after being adjusted as above, may not be embeddable in two or even three dimensional space. Thus, we must use an MDU algorithm which will find an optimum fitting of the distances when an exact solution does not exist.

Two MDU algorithms were evaluated for uLG-1. The first was from the R package, `munfold` which uses the well-known algorithm of Schönemann (1970) [57]. Since Schönemann's algorithm, itself, uses linear algebra to compute exact solutions, but does not find optimum, non-exact solutions, `munfold` refines Schönemann's solution using conjugate gradient methods. Distances are first transformed to allow Schönemann's linear algebra operations to compute a solution before optimization is attempted. Perhaps because of the often large distances used to model zero or near-zero probabilities, this method, while quite good at finding exact solutions, had considerable difficulty with uLG-1 data. The second algorithm considered, and ultimately chosen for the uLG-1 algorithm, was `smacofRect` from the R package `smacof` of De Leeuw & Mair (2009) [24]. This algorithm uses majorization to minimize unfolding stress (SMACOF stands for Scaling by Majorizing a Complicated Function). It seems to have little difficulty dealing with the large distances, and sometimes large category/object ratio, of uLG-1 data. Our uLG-1 algorithm, known as `UnbiasLG`, was coded in R and is found in the supplementary file. As arguments, it takes a list of indicator matrices, a maximum number of iterations, a chosen dimension for the configuration, and a desired APWL.[9] Outputs include coordinates, APWL, and variable distance matrices for the configuration.[10]

At this point, it is instructive to look at two basic demonstrations. For the first, we shall analyze compositional data on alcohol consumption compiled by the

[9] APWL rather than fixed-coordinate precision is used to gauge convergence of the algorithm to avoid the possibility of the algorithm reflecting, rotating, or translating from one iteration to the next. If this should occur, the algorithm could find a series of optimum configurations, but fail to converge due to relatively large changes in the coordinates found. In practice, this does not seem to happen, but it is certainly a theoretical possibility.

[10] It should be noted here that loss in the uLG-1 algorithm comes from two sources: First, from the fact that, at each iteration, we are minimizing a majorization, not the deviance itself and, second and more important, from the fitting of non-embeddable distances by MDU. A method to minimize this latter source suggests itself. As we have seen in discussing negative target distances, adding a constant to all within-variable distances results in equal model fit (in terms of APWL). What if, for our chosen dimension, we could compute a constant at each iteration that, by adding it to the distance matrix, would produce embeddable distances. MDU (or distance-fitting) loss would then be zero. This is a version of a well-known problem in MDS, the additive constant problem. It has been shown that a constant can always be found such that the distances can be embedded in (n-2)-dimensional space, where n is the number of objects. In fact, in Calliez (1983) [8] a formula is derived to find the smallest such constant. We have tried to use this constant in the above approach without success in improving the performance of the algorithm. Finding such a constant for a fixed dimension (usually two or three) less than (n-2), a problem apparently first proposed by Torgerson (1952) [64], cannot always be done. One must settle for a constant that will give best fitting distances in a least-squares sense. Finding this involves something of an elaborate heuristic process or an iterative approximation. See Messick & Abelson (1956) [48] and Cooper (1972) [10], respectively. See also De Leeuw & Heiser (1982) [22]. Computing this constant and using it to transform the $-\tilde{z}_{ijl}$ matrix at each iteration of the uLG-1 algorithm may be worthwhile for some applications, but our experimentation with this approach has found it not to yield greatly improved efficiency or performance, particularly in comparison to other methods that will be reported on.

World Health Organization (WHO). (For all data referred to herein not appearing in the text, including this dataset, see the supplementary file.) From an original dataset with 95 countries[11], 47 were chosen at random. The table shows percentages for each country indicating that country's preference among the four listed types of alcoholic beverage. For our first analysis, we wish to use LG to perform a cross-classification of the countries by continent versus whether or not beer is the preferred drink. Thus, we will use two indicator matrices, one of dimension 47 × 2 with categories for beer and other and one of dimension 47 × 4 with categories for Africa, Asia, Europe, and North/South America.

The `UnbiasLG` algorithm was run on this data, with initial coordinates in 2 dimensions, until an average point-wise loss (APWL) $< .001$ was achieved. (Recall that average point-wise loss (APWL) means the average of the absolute differences between data probability and corresponding model probability.) The resulting plot of objects (countries) and categories is shown in Figure 1.1.

Notice that there are 8 possible cross-classifications of the countries, but only 7 clusters of objects produced. A quick check of the data shows that all of the NA/SA countries preferred beer; i.e., no country in the list has an other-NA/SA profile. Further, it is important to note that the 47 objects are each assigned to one of 7 points, depending upon which of the existing dual profiles they have. (Points in the plot were jittered to provide legibility.) Thus, LG provided something of an exact clustering of this data. We see that, for most countries, regardless of continent, beer is the preferred drink. Those countries with a different preference were separated from their dominant continent groupings and pulled to the other category point in nearly radial directions with the beer category near the center point. The sub-groupings contain countries all from the same continent with their

[11] In the original data, Poland's percentages were entered erroneously. The Poland row totaled only .52. The row was removed and the evenly numbered rows of the remaining 94 × 4 table were used for the analysis.

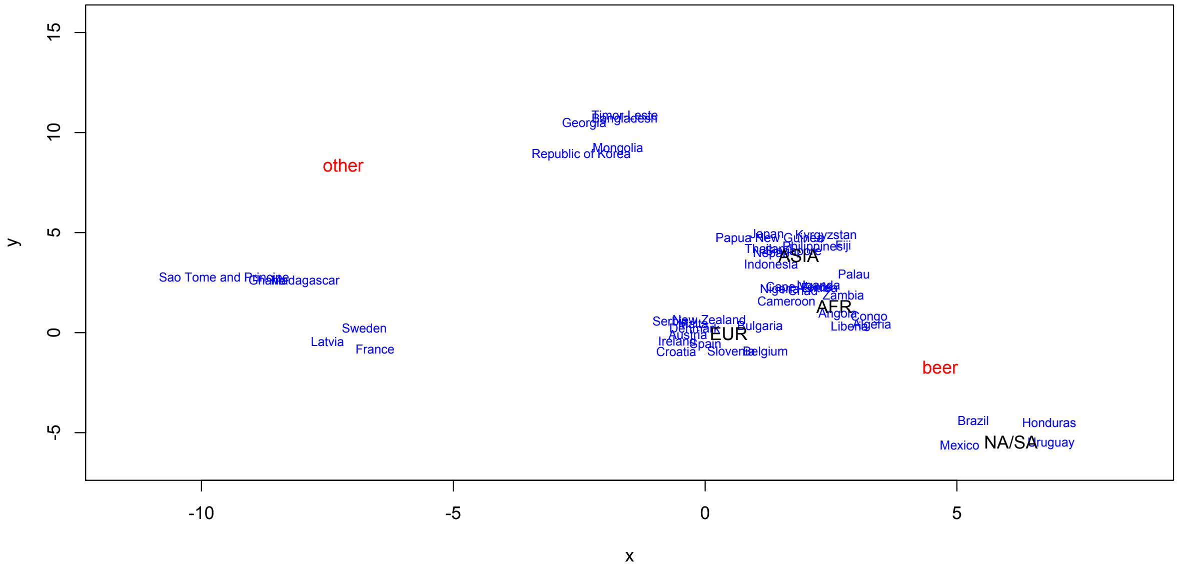

Figure 1.1: WHO Beverage Cross Classification

respective main continent groupings between them and the beer category point. Notice that they are closest to their respective main continent groups.

In addition to this clustering process, recall that distances between objects and categories (and, for this cross-classification, profile clusters and categories) determine model probabilities. Working with this sort of cross-classification involving binary indicator matrices, these probabilities are, again, somewhat conceptual. They provide us with our measure of model fit, of course, but are not necessarily of primary importance in the data analysis. When working with fuzzy-coding applications, they take on more significance. We turn to these next.

Recall that a fuzzy-coded indicator is a non-binary, row-stochastic matrix; i.e., all entries are non-negative with at least one row having at least two positive entries and a row sum of 1. Some rows (often, all or nearly all rows) of the indicator matrix (or object profile) are, therefore, non-degenerate discrete probability vectors. To better understand the ULG-1 model applied to such data, we consider, as above, some basic examples.

First, suppose that row i is uniform over m categories; i.e., the probability that object i is classified in any of the categories is $1/m$. Then, it is easy to see that, for an exact model, all category points should be placed equidistant from the object i point - on a circle if we are in 2-dimensional space (or in general, a d-dimensional sphere). Of course, if we are plotting only object i, then all category points can be placed at any identical point (even the object point) since, from the probabilistic viewpoint, the categories are identical. Thus, as with binary indicators, without constraints, we have unidentifiable models. Suppose then that, in addition to object i, we have several other objects each with differing profiles. It will of course no longer be possible to plot all categories at a single point. In trying to reposition them, however, we will try to keep the categories on some sphere about object i, the uniform profile, then position other objects according to the model equations. This, in essence, is what the LG algorithm carries out.

Note the following basic properties of LG with fuzzy coding:

For any $n \times 2$ fuzzy indicator with all positive entries, there is an exact 1- dimensional solution. In fact, there are infinitely many different (i.e., non-equivalent) solutions. For m (the number of categories) > 2, notice that if one row profile is uniform and a second is not, a configuration of at least dimension 2 is necessary for perfect fit (which is not always attainable). This has implications for missing data applications and for psychometric theory.

Also, as shown in (29), the absolute difference in distances from an object point to 2 categories gives the model log-risk of the object being in the closer (more probable) versus the farther category. This is an aspect of the model which provides a connection to multinomial regression, as we shall see.

We close this introductory section with an application using fuzzy coding. The data is the same WHO alcoholic beverage data analyzed above. This time, we will consider only one variable, the preference percentages as compositional data. We can apply the `UnbiasLG` algorithm directly to the table of percentages. Figure 1.2 shows the resulting plot after running the algorithm until an APWL $<$ .01 is attained.

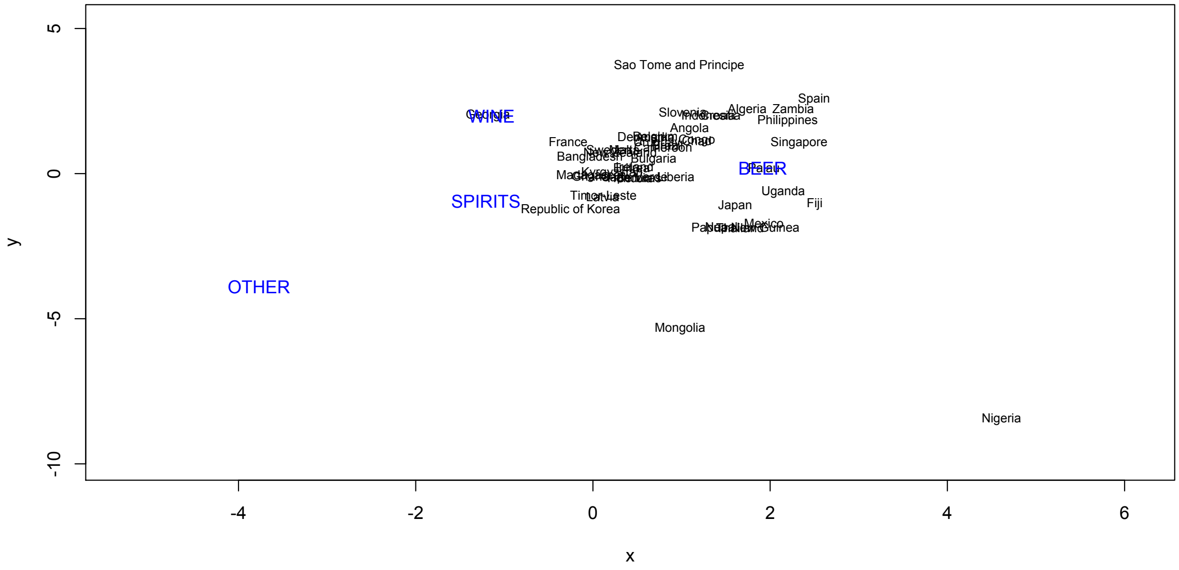

Figure 1.2: WHO Beverage- Fuzzy Indicator

Notice that the OTHER category, which has several 0 entries, is pushed away from the rest of the category points and the region of object points. Mongolia and Nigeria, countries with by far the largest percentages favoring an OTHER beverage (.385 and .350, respectively), are positioned to reflect that. Both have a lack of preference for wine which is reflected as well.

By way of basic interpretation, we note that Georgia, Sao Tome, and France are the countries most favoring wine, with Korea, Timor-Leste, and Latvia favoring spirits. The group of countries including Philippines, Algeria, Zambia and Spain are countries in which beer and wine together make up nearly 100% of beverage preference, with both categories having some non-negligible percentage. A large group of countries (which include Sweden, Bangladesh, New Zealand, and Papua New Guinea) have close to uniform density among beer, wine, and spirits. Such a profile should be located near the center of the unique circle containing these category points; or, equivalently, at the intersection of the bisectors of segments joining them. (It is worth repeating here that when interpreting an LG plot it is important to keep in mind three ideas. First, relative distances in terms of absolute difference (not ratio) are of interest, second, due to the exponentiation used in the model, small apparent differences in distance can indicate large differences in the model probabilities, and third, and perhaps most important, given non-trivial data, objects with similar profiles are positioned close to each other, as in CA.)

Notice that BEER, SPIRITS, and OTHER appear nearly co-linear. If they were exactly so, this would indicate that, in 2 dimensions, there is no country that can be plotted to have exactly uniform preference for these categories. Put another way, the closer three category points are to being co-linear, the farther away from all three will be the unique point (in 2 dimensions) equidistant to them. For three co-linear categories, there is no such point. Thus, to plot an object with relatively equal distances to the three categories, we must plot it relatively far from all three, and the farther, the better, in terms of precision. (The converse notion to this line of thought is that, in theory, absent such an object in the dataset, the LG algorithm can plot the corresponding categories in close to a 1-dimensional representation.) For our data, Mongolia is the only candidate for such a point; i.e., it is the only country with even close to uniform preference for these beverages (0.200, 0.370, 0.385).

These comments apply as well to the OTHER, SPIRITS, WINE categories. Notice that these points lie on a very broad arc. The circle containing this arc has its center far from the region of country points in our plot. Thus, a country plotted there, having uniform preference for those three categories of beverage (with a relatively small preference for beer), would be quite unusual. There is, in fact, no such country in our data. It should be noted that such a partially uniform profile could be approximated quite closely by plotting a country point far from the three categories in a direction determined by the country's preference for beer. Mongolia and Nigeria should be examined in this regard. Notice that neither has close to uniform preference among the OTHER-SPIRITS-WINE categories (Mongolia: 0.200 (beer), 0.385 (other), 0.370 (spirits), 0.045 (wine)) and Nigeria: (0.500 (beer), 0.350 (other), 0.100 (spirits), .050 (wine)).[12] Their respective positioning relative to the BEER category is explained by their beer preferences.

1.2 Voronoi Cells

If applying the uLG-1 formula results in a perfectly-fitted model, the distances between object points and category points will be ordered in the same way as the probabilities with which the objects are placed in the corresponding categories. That is, each object will be closest to the categories with which it is placed with highest probability, second closest to the category with second highest probability, etc. Thus, along with the probabilistic metric discussed above, uLG-1 also provides a non-metric or ordinal model of the data. A useful geometric construct for studying this aspect of the LG output is the Voronoi cell.[13]

[12] The model probabilities for Mongolia are : 0.202 (beer), 0.371 (other), 0.409 (spirits) 0.018 (wine) and for Nigeria: 0.427 (beer), 0.258 (other), 0.287 (spirits), 0.028 (wine). The results for Mongolia are modeled quite accurately while Nigeria's relatively strong preferences for the BEER and OTHER categories made positioning it more difficult. This suggests interesting properties of model fit to be examined below.

[13] After this brief introductory section, we will return to the subject of Voronoi cells in our discussion of the convergence of LG algorithms. For our purposes, we need not explore the mathematics of Voronoi diagrams in great detail, though that certainly has been done in the field of computational geometry. The interested reader is referred to Aurenhammer (1991) [4] for a concise introduction of the subject with a thorough bibliography.

The Voronoi cell of a category point is the set of all points in the model space that are closer to that category point than to any other category point. This set of points depends, of course, upon the configuration of all category points in the model. Notice that the IMT of De Leeuw can be stated in terms of Voronoi cells as follows: For any $\varepsilon > 0$, object and category coordinates can be found by the LG algorithm such that the model loss is less than ε if and only if, for each variable, each object can be placed in the Voronoi cell of the category to which it belongs.

Again, an example is instructive for studying Voronoi cells. Figure 1.3 is a 2-dimensional LG model (with the Euclidean distance metric) for a subset of the WHO beverage data. The data for this model was obtained by removing from the data used above all six rows in which the OTHER category was non-zero. This left 41 countries. Then the OTHER category was removed leaving a dataset with dimension 41 × 3. On this data, our LG algorithm was run in R, this time until an APWL $< .001$ was attained.[14]

It is clear that, using the Euclidean distance model in 2 dimensions, to find the Voronoi cells of this plot we simply draw the perpendicular bisectors of the segments joining the pairs of category points. For three categories, there are 3 such bisectors and, if the category points are not collinear, these will intersect at a unique point, as they have in Figure 1.3, that point being the center of the unique circle through all 3 points. Each bisector divides the plane in half. The Voronoi cell for each category can easily be seen as the intersection of the 2 half-planes

[14] Here, it is instructive to consider another property of uLG-1. The APWL of this 41 country model was, in fact, .00099, attained after 499 iterations of the algorithm. Most probabilities are fit exactly. In this dataset, there are 3 (out of 123) zero probabilities. In a simple experiment, we removed these rows leaving us with a 38 × 3 dataset. An APWL of .00099 was reached for this dataset after only 194 iterations and the APWL for the uLG-1 model of this dataset after 499 iterations is .00052, nearly 50% less. For the 41 country data, APWL was .00197 after 499 iterations. Notice how much of the loss in the 41 object model (nearly half) was due to fitting the three 0's (out of 123 points) using finite distances and the increase in efficiency of the algorithm resulting from their removal. This is a common characteristic of uLG-1models and useful to keep in mind for model interpretation and further study of the algorithm.

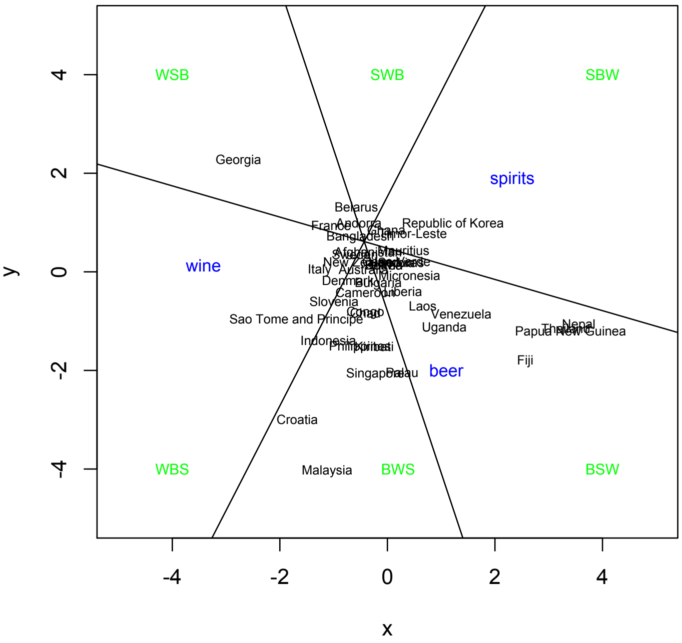

Figure 1.3: WHO Beverage - Voronoi Cells

containing the category point created by the bisectors involving that category or, equivalently, the union of two regions known as *cones* emanating from the common point of intersection of the bisectors.

Here, it is helpful for fuller understanding of LG and Voronoi cells to undertake a basic analysis of our model. The most obvious feature to notice is that Bangladesh is plotted nearly right at the intersection of the bisectors. As discussed above, we therefore expect Bangladesh to have a uniform preference for the 3 beverages. In fact, its preferences are 0.305 (beer), 0.362 (wine), 0.333 (spirits) and it is easily the country with closest to a uniform preference.[15] Its model results are identical to the data. Next, we see that Sao Tome, Georgia, Italy, and France are clearly within the WINE Voronoi cell. They are the four countries with the highest preference for wine (0.687, 0.924, .650 and 0.590 respectively). Notice, however, that of these four, Georgia is furthest from the wine category despite having by far the highest preference. We are tempted to wonder if there are problems with fitting these points, but model fit here is optimum: for France we have 0.190, 0.590, 0.220 for both data and model and for Sao Tome, 0.275, 0.687, 0.038, for Italy .25, .65, .1 and for Georgia .023, .923, and .054. There are several other pairs of points that demonstrate this same pattern. The analysis of these points highlights the operative principal, commented on previously, in analyzing an LG plot as a metric model; namely, it is relative distance differences

[15] As noted above, the presence of such an object profile in data with many other objects with profiles that greatly diverge from the uniform requires at least a 2-dimensional model for optimum fit. In fact, attempting to fit a 2-dimensional model with a 1-dimensional initial category configuration resulted in poor model fit even running the uLG-1 algorithm with nearly twice as many iterations. Thus, we see that, in LG, the initial configuration can effect model fit just as in MDS. In a thorough study of this issue by Spence (1972) [60], he surmised that this is due to local minima being found from some starting points. For uLG-1, we have found that, in general, a 2-dimensional, concentric circle initial configuration seems to work well, even for 1 and 3 dimensional models; i.e., at least as well as more strategically devised starting configurations. For example, using the HOMALS configuration as the initial configuration seemed to be a promising approach. HOMALS seeks to place objects closest to the categories they are in and the IMT would suggest that this would lead to efficiency gains in running uLG-1. Unfortunately, we did not obtain greater efficiency nor greater precision than with the concentric circle starting configuration. See Gifi (1990) [33] and Michaelidis & De Leeuw (1998) [49] for a complete description of the HOMALS approach.

that are important. Thus, often we must combine this metric approach with the ordinal approach using Voronoi cells to fully analyze the models.

Looking further at Figure 1.3, notice that the three Voronoi cells are each unbounded. They are each composed of two of the six unbounded regions created by the bisectors. It is easy to see that these regions correspond to an ordering of beverage preferences among the country points that fall within them and they are labelled with the orderings to which they correspond. In MDS, these are referred to as *isotonic regions* associated with the orderings. In Figure 3, there are six such regions, of course, each corresponding to one of the six possible orderings of the three beverages. Based on the model, we would expect no country in the dataset to have a clear preference for spirits, wine, and beer (SWB) in that order, which is, in fact, the case.

Figure 1.3 nicely illustrates a classic result attributed to Coombs (1964) [9] that if there are *n* category points in *n-1* dimensional space, then the isotonic regions are unbounded. Thus, given the flexibility available for locating points into the proper unbounded isotonic region, for 3 categories we can expect a 2-dimensional LG model to fit quite well in terms of APWL. This is the case here in which a very satisfactory model fit was achieved in a relatively small number of iterations. If the number of category points is increased, then, in general, more bounded isotonic regions will result. In connection with multidimensional unfolding, it is shown by Borg & Groenen (2005) [6] that, even with the constraining effects of these bounded regions, satisfactory models will still be found if the objects are such that they are relatively evenly distributed throughout the order relationships among categories. As we discovered above, our data here is not of this sort. Nearly all of the countries have a leading preference for beer, none fall in the SWB region, as previously noted, and very few in either the WINE or SPIRITS cells. We would expect, then, that supplementing this data with additional objects requiring additional categories would make such a model fit more difficult to attain. In fact, we saw this in our model of the 47-country, 4-category dataset. In that model, our LG algorithm produced 10 times the APWL and a model that had some difficulty fitting one of the supplementary objects (Nigeria).

CHAPTER 2

Applications of uLG-1

2.1 LG and MDU

LG being a new, largely untried method, it is useful to begin by presenting some applications in established data analysis settings. A natural starting point for considering applications is to examine how LG can be used to carry out standard MDU. We shall apply it to the `breakfast` dataset from Green & Rao (1972) [37]. Forty-two individuals were asked to rate fifteen breakfast items from 1 (favorite) to 15 (least favorite). Further information and a classic MDU analysis of the dataset can be found in De Leeuw & Mair (2009) [24] and the `smacof` package in `R`.

The classic MDU approach is to consider the ratings as distances then use `smacofRect` to compute rater (object) and item (category) coordinates. The LG approach is slightly different. We must first transform these distances to probabilities in some sensible way to create a fuzzy indicator matrix. A logical way to do this is to use the LG link with the ratings as the distance inputs.

This approach results in the plot shown in Figure 2.1. The APWL is .028 with 41 of the 42 raters properly classified (by being closest to their favorite item). Interestingly, the configuration is somewhat similar to the `smacof` plot. Raters are mostly centrally located. The items cluster into *toast*, *muffin*, *cake*, and *sweet* groups with raters who have strong first preferences for these positioned near the

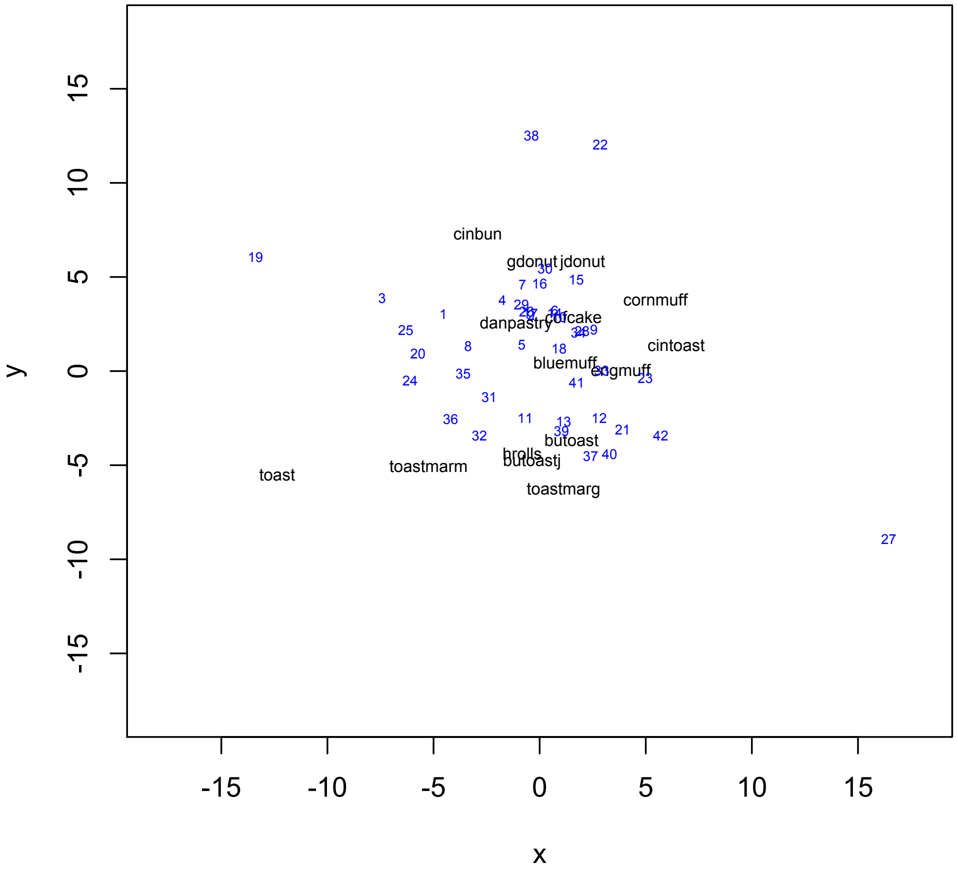

Figure 2.1: Breakfast with LG Link Probs.

clusters. The only noteworthy difference is that, here, `danpastry` and `cinbun` are switched with `danpastry` grouped as a *cake* and `cinbun` as a *sweet.* In the `smacof` plot, the opposite is true.

Of course, there are other straightforward ways to transform the ratings into probabilities. An obvious one is simply to use a uniform latent variable approach in which a rating of 1 corresponds to a probability of $\frac{15}{120}$, 2 to $\frac{14}{120}$, etc. Using this as an indicator, uLG-1 produces the plot in Figure 2.2. Here, we see a somewhat different configuration, but one that, nonetheless, reveals similar structure. The *toast* cluster is presented as qualitatively different from the rest of the items and the raters with *toast* preference (32, 36, 37, & 39) are positioned nearby. The remaining items are arranged on a scale from *sweet* to *cake* to *muffin* with raters appearing to be similarly aligned. The `danpastry` and `cinbun` are more closely associated and positioned closest to *cake.* APWL is .018 and, on their favorite item, only 15 of the 42 raters are correctly classified. It appears, however, that, overall classification, taking into account second, third and least favorites, may be more accurate than the LG link plot.

That such different configurations can reveal similar structural relationships in the data is something that, as we shall see, must always be kept in mind when applying LG. Here, the difference arises from the different transformations used. Figure 2.3 shows how the two transformations we used differ in their scaling of the rated items. (The straight line is, of course, the uniform transformation.) Selecting the most appropriate transformation is a task that will depend on the extrinsic knowledge of the researcher. As illustrated here, the affect of a particular transformation must always be examined and accounted for.

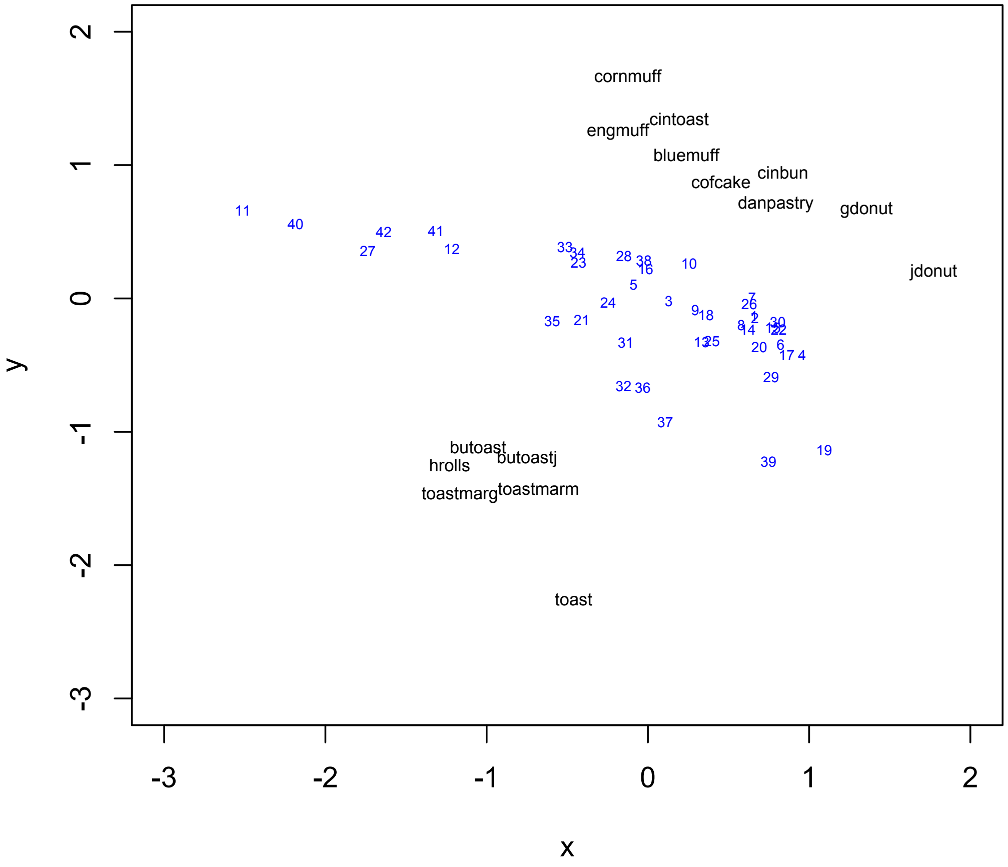

Figure 2.2: Breakfast with Uniform Probs.

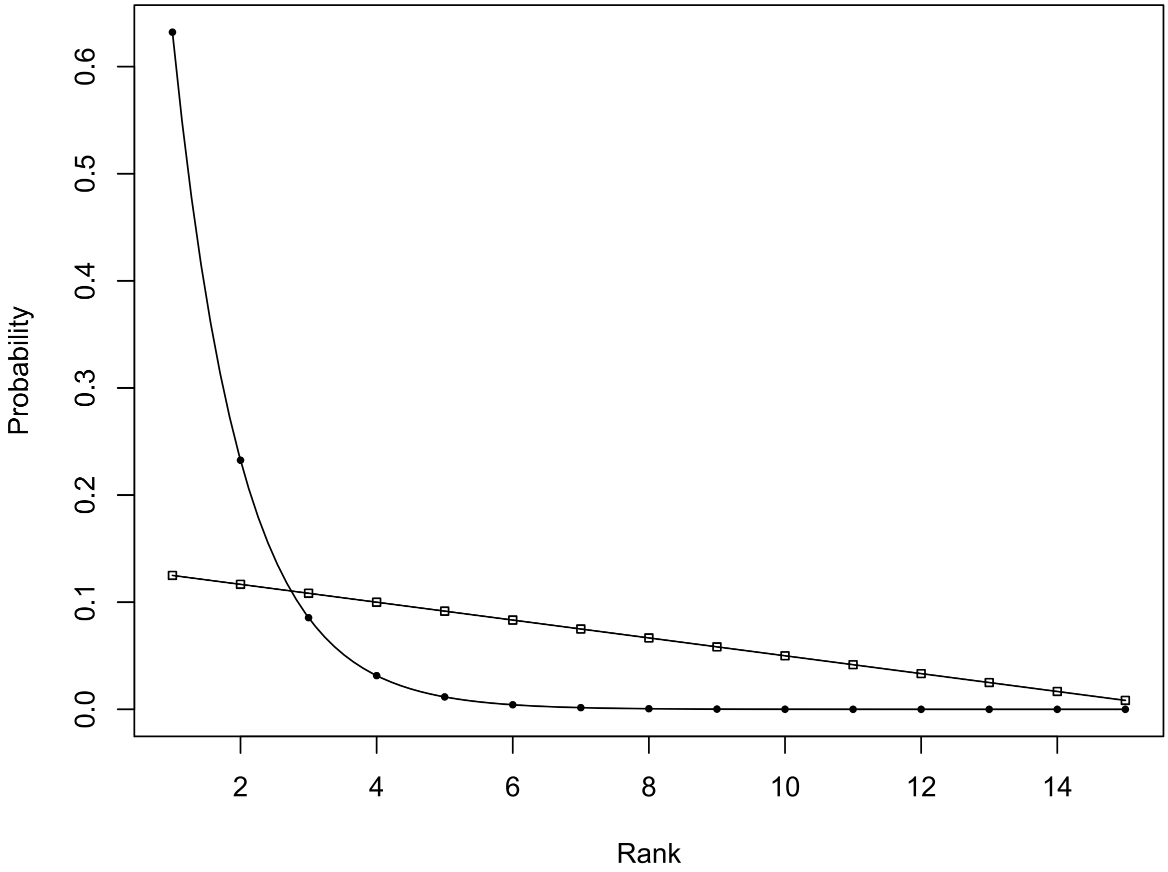

Figure 2.3: Breakfast Transformations

2.2 Roll Call Data

Historically, the study of choice behavior has been perhaps the primary source of distance association models. We have already referenced Shepard's model, for example, as an ancestor of LG. Ranking data, like the `breakfast` dataset, is a special case of this type of data, but perhaps the most fundamental type is voting, or roll call, data.

Distance methods have been used for many years to study this type of data. In its most basic form, voters, typically legislators, are the objects and Yea or Nay votes are recorded for them on a number of bills or resolutions. In the context of LG, the data is expressed by a 2-column binary indicator for each vote. Over the years, several analysts have developed scaling methods for such data. As with LG, the aim of these methods is to locate voters and choices in a low-dimensional (1 or 2) psychological space in such a way that the voters are closest to all their choices. In 1-space, this means voters are on the correct side of the midpoint of the choice points; in 2-space, of the perpendicular bisector of the segment joining the choices. Of course, when many votes are involved, 100% classification is generally impossible so a loss-function-minimizing approach is needed.

Currently, WNOMINATE of Poole et al. (2011) [52] appears to be the most popular method for this type of modeling. It extends the NOMINATE method of Poole & Rosenthal (1985) [53] to allow for scaling in more than 1 dimension and to account for polytomous voting (i.e., abstentions and absences).

The WNOMINATE model is somewhat elaborate. It postulates a utility function for each legislator's Yea vote on each bill as:

$$U_{ijy} = u_{ijy} + \epsilon_{ijy}, \tag{30}$$

where:

$$(31) \qquad u_{ijy} = \beta \text{ exp}(\text{-}\frac{\sum_k^s w_k^2 d_{ijyk}^2}{2}).$$

Legislators are assumed to maximize their utility with their votes, subject to some random error. In the above, d_{ijyk} is the distance between a legislator's ideal point and a particular Yea vote point in an s-dimensional space, β and the w's are types of bias parameters, and the ϵ's represent an error term which is assumed to follow an extreme value distribution.

Given this distribution assumption, the probability that legislator i votes Yea on bill j is:

$$(32) \qquad P_{ijy} = \frac{exp(U_{ijy})}{exp(U_{ijy})+exp(U_{ijn})}.$$

In the manner of LG, these probabilities are used to construct a likelihood function and parameters are found to maximize this function. A block-relaxation algorithm is used to do this. First, vote point coordinates and are found by fixing voter points and bias parameters. The vote points are reported by their midpoint and the distance between them. Next, with vote midpoints and between-vote distances fixed from the first step, voter points are found. This involves using SVD to solve a linear least squares problem. Compare to De Leeuw (2006) [18] for a similar approach. Then, the bias parameters are found, subject to heuristically determined constraints. This process is iterated until all parameters correlate at 0.99 or better with the previous iterations's estimates.

In this study, we will compare the more parsimonious LG approach, which can also represent the data in any dimension and for any number of vote choices, to WNOMINATE on two datasets found in the `R` package `wnominate` of Poole et al. (2011) [52]. The first dataset records the votes for three sessions of the 59-

country UN Security Council. A total of 237 votes are recorded with Yeas, Nays and abstentions. However, to achieve greater stability in its stochastic modeling, wnominate removes near-unanimous votes (18, in this case). These are votes in which the losing side has fewer than 2.5% of the vote. Since this occurs usually with procedural or ceremonial votes that do not reflect the ideological or psychological dynamics of the voting process, we consider this a sound approach and have followed it in our uLG-1 model. Using its SVD approach, wnominate determines the lowest adequately-fitting dimension for the choice space - 2 in this case - and finds coordinates. We therefore have run uLG-1 for a 2-dimensional plot. It is shown in Figure 2.4. The obvious division is between Warsaw Pact and non-Warsaw Pact countries and is nearly identical to what is given by wnominate. Some other interesting country groupings (geographical and political) are visible as well. Figure 2.5 shows LG's classification rates across the 219 votes. Overall classification is 78.2%, comparable to wnominate. This rate includes abstentions and, since there are relatively few of these, they are not likely to be correctly classified. Also, our overall rate is influenced by 2 outlier votes. It would most likely be of interest to a political analyst to examine these votes more closely.

The second dataset from wnominate records 596 votes of the 90th Senate of the United States. This covers the last years of the Johnson administration, 1967-1969. We will compare the 181 agriculture-related votes to the 338 non-agricultural related votes[1], following a study suggestion in Poole et al. and the identification, in the paper, of the agriculture vote numbers. Again, we will use a two-dimensional plot since wnominate found this to be adequate for modeling the entire voting dataset.

Our first plot, shown in Figure 2.6, reveals an interesting feature of the data. One Republican and one Democrat are moved fairly far from the remaining objects

[1] There were 27 near-unanimous agriculture votes and 50 near unanimous non-agriculture votes that were removed, following the procedure noted above.

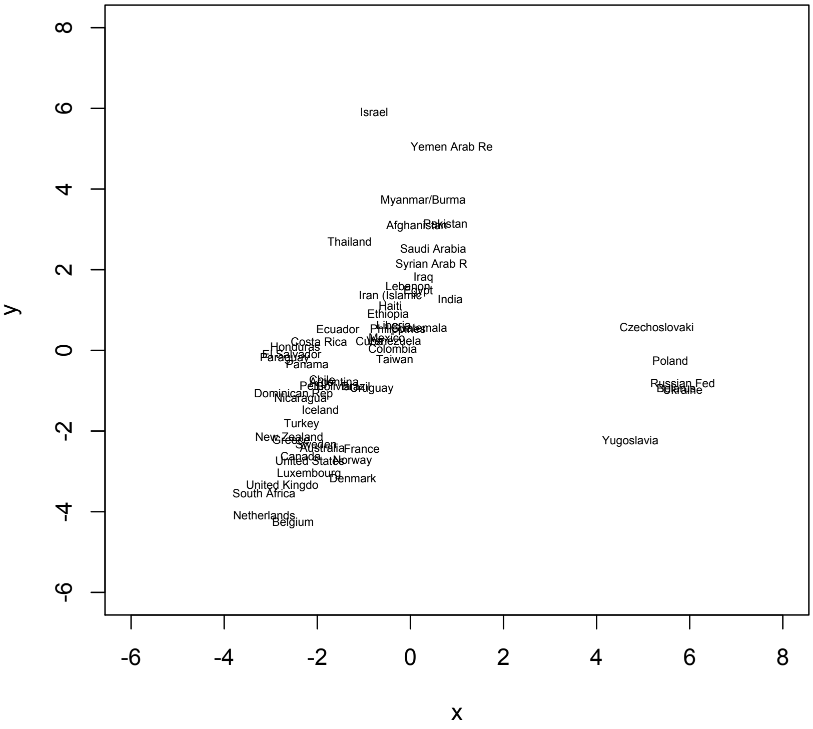

Figure 2.4: UN Voting

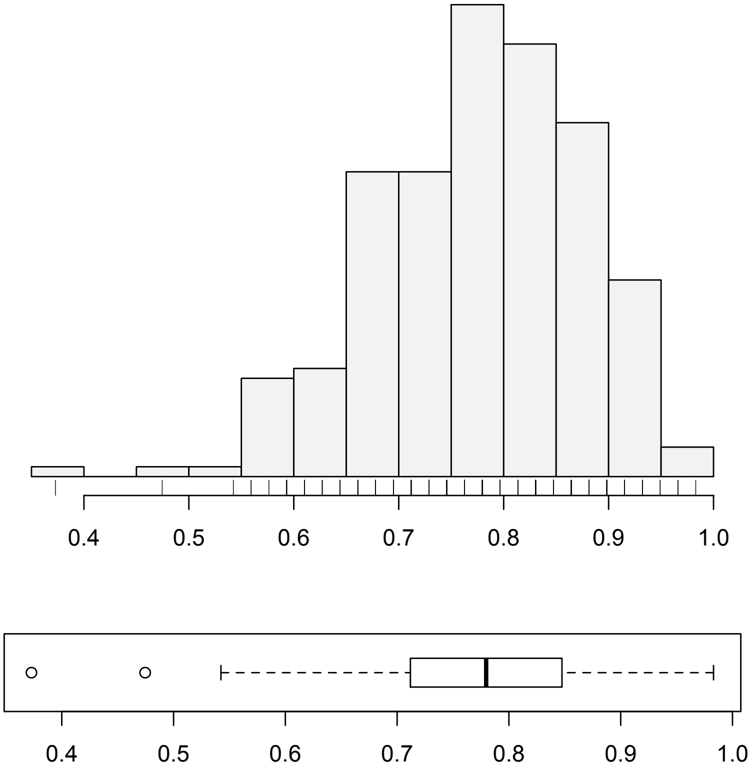

Figure 2.5: UN Classification Rates

in their parties. The Republican is Goodell and the Democrat is Robert Kennedy, both of New York. Goodell replaced Kennedy after the latter was assassinated in June, 1968. Thus, both men were absent for portions of the Senate votes. These absences are recorded as a fourth category of voting and they are the only 2 voters with any 1's in that category. Thus, they are regarded by LG as somewhat similar, even though it is very well known that their actual votes place them at the opposite ends of the political spectrum in the United States at the time. This gives an interesting look at the possible uses of LG for examining missing data, since the absences can be so regarded. For our purposes here, however, we decided to remove these 2 voters since their effect on the data can be fairly well accounted for.

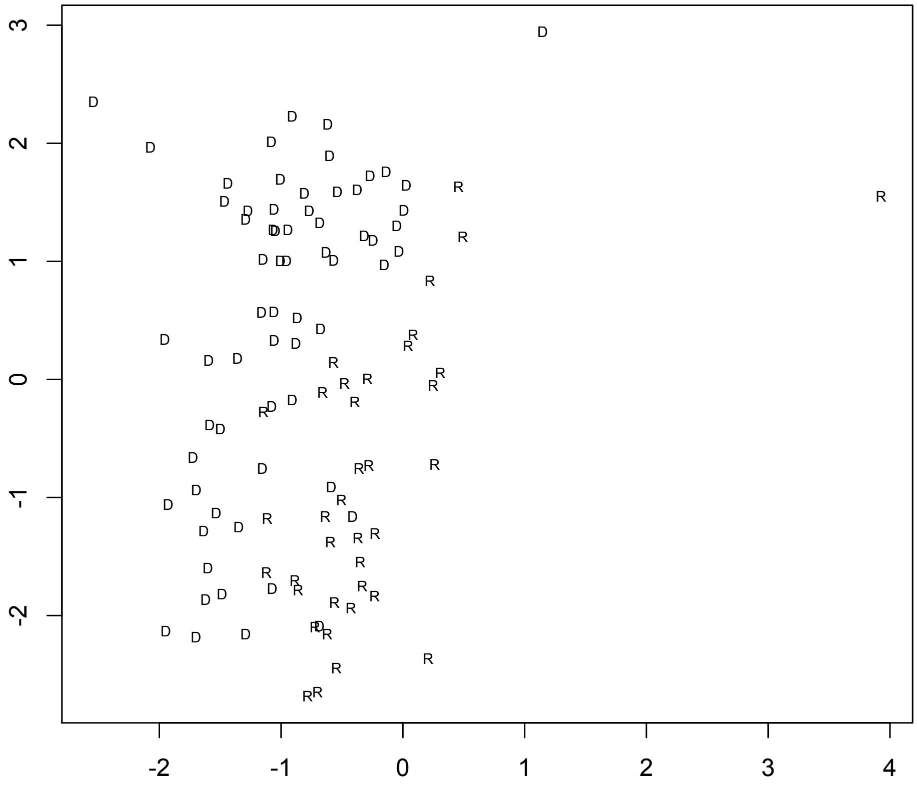

Figure 2.6: 90th Senate - Full Senate Agriculture Voting

The resulting plot is shown in Figure 2.7. It is a fairly similar configuration (with a rotation), perhaps giving slightly clearer demarcation of the parties, but showing, as in Figure 3, some overlap and a tendency, particularly among some Democrats, to move to the middle on agriculture issues. Figure 2.8 shows the classification rates. Very few votes are below 60%, none under 50%, and the overall rate is 75.8%.

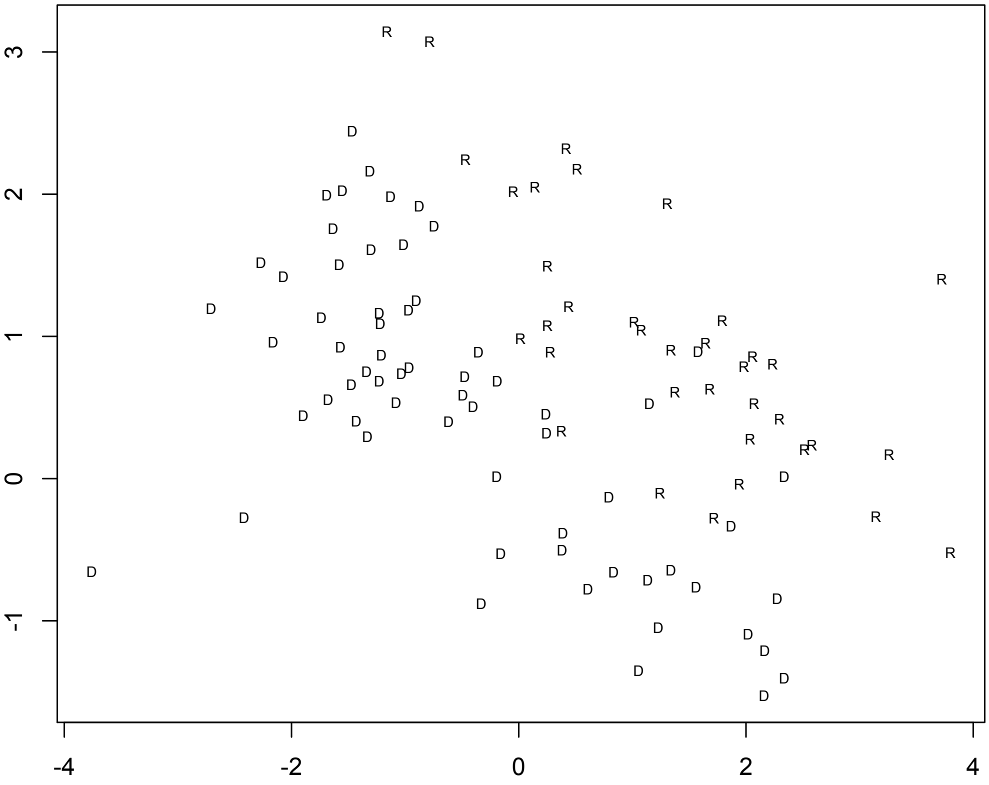

Figure 2.7: 90th Senate - Agriculture Voting with NY Senators Removed

Next, we plot the remaining Senate votes, again removing Goodell and Kennedy. The configuration is shown in Figure 2.9. Here, we see much clearer demarcation of the parties, indicating a higher degree of partisanship on the non-agriculture related votes. The plot is quite similar to the one obtained by `wnominate` shown in Poole et al. (2011) [52]. Figure 2.10 shows the classifications. Even with two

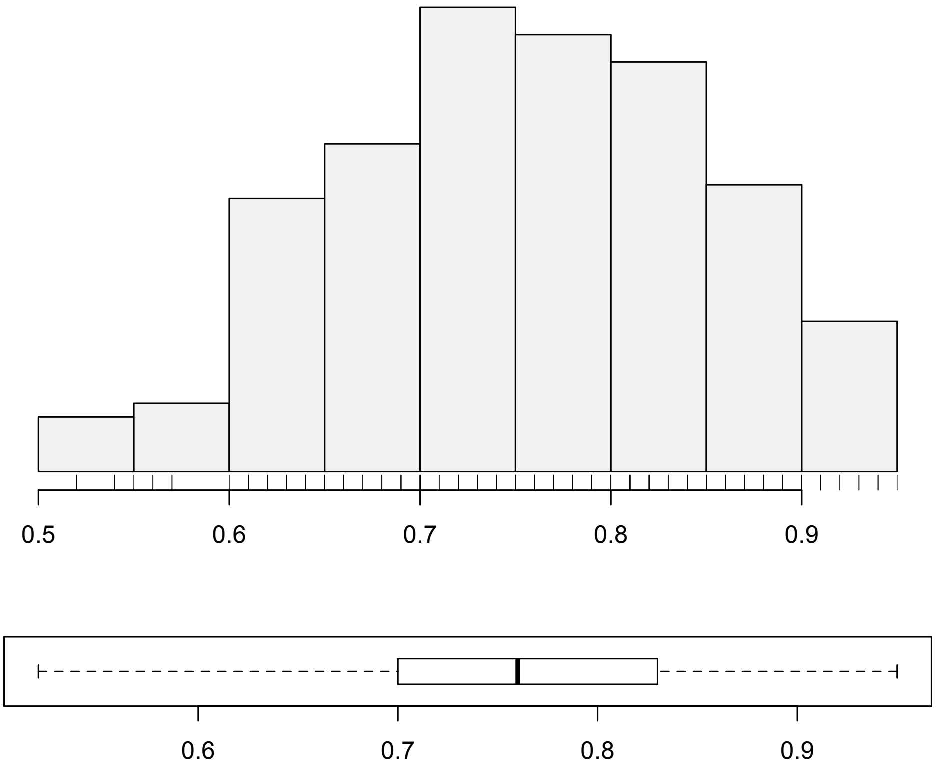

Figure 2.8: 90th Senate - Agriculture Voting Classification Rates

votes below 50% classification (which, again, would likely be of interest to an analyst), overall classification is 76%, comparable to the `wnominate` rate for the entire dataset.

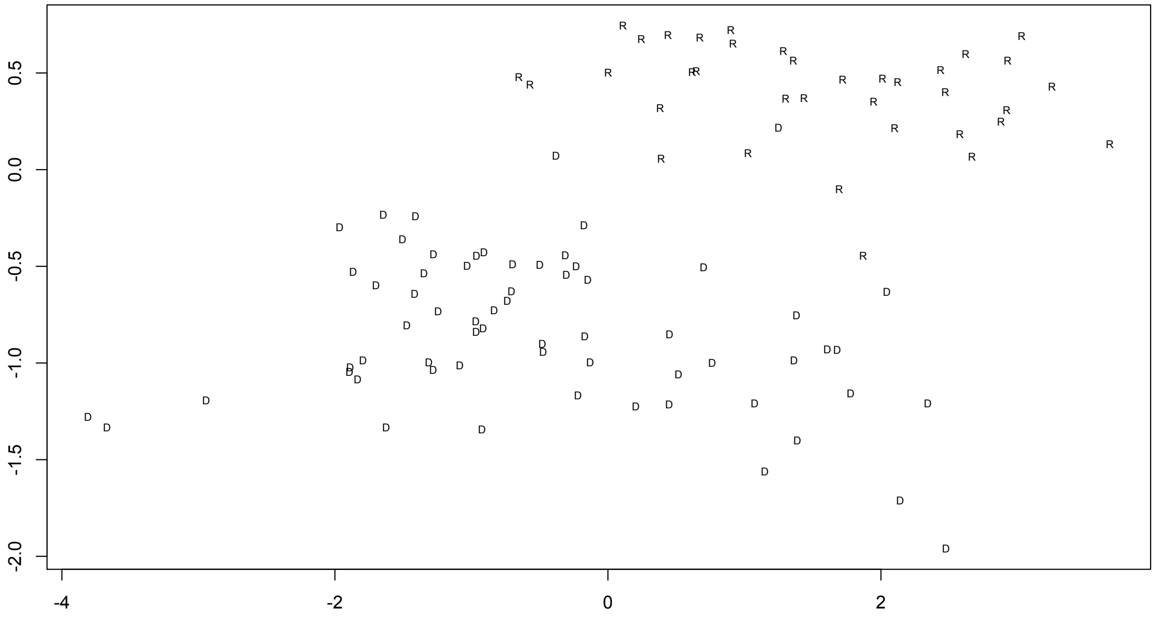

Figure 2.9: 90th Senate - Non-Agriculture Voting with NY Senators Removed

2.3 Multinomial Regression

We next apply LG in connection with multinomial regression. In multinomial regression, a response variable with J $>$ 2 categories is modeled as a linear system of explanatory variables. The explanatory variables can be either discrete or continuous and the response categories can be either nominal or ordinal. As with binomial (logistic) regression, a logit model must be used to link the category probabilities to the linear combinations of explanatory variables to ensure that the predicted probabilities are between 0 and 1.

For nominal response data, the general multinomial logit model is used. (It can be used for ordinal data as well, but the information about order is not used

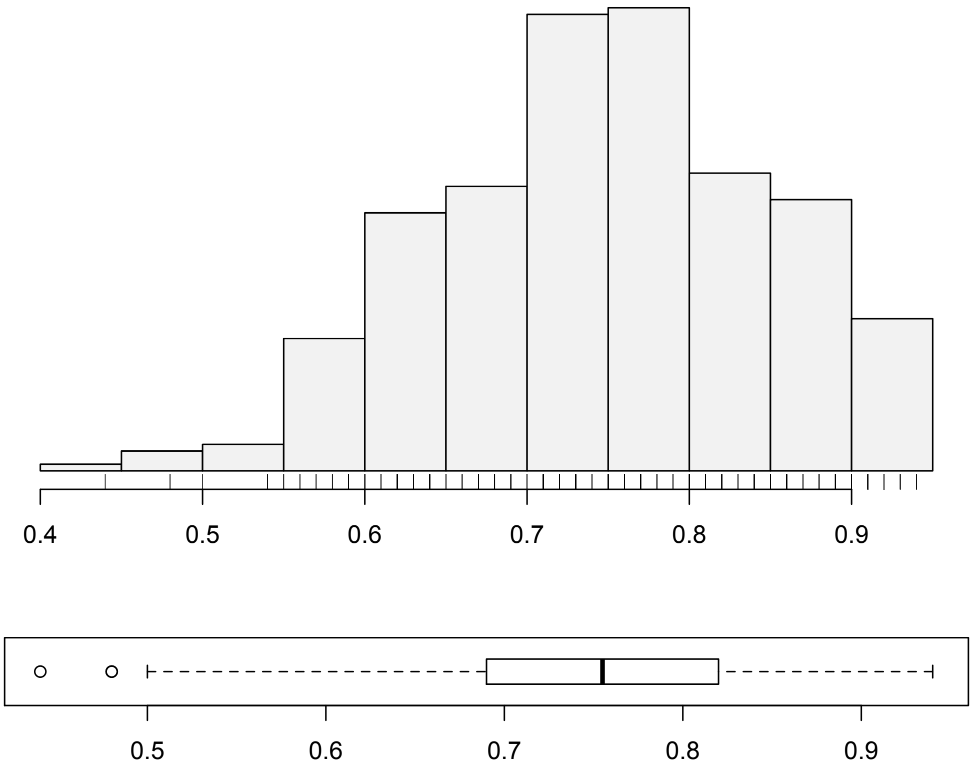

Figure 2.10: 90th Senate - Non-Agriculture Voting Classification Rates

resulting in some loss of model fit.) See Faraway, (2006) [28]. The general logit model is:

$$(33) \qquad \mathrm{p}_{ij} = \frac{exp(\beta_j x_i)}{\sum_{k=1}^{J} exp(\beta_k x_i)}.$$

For identifiability, we set $\beta_1 = 0$, so we have $\mathrm{p}_{i1} = 1 - \sum_{j=2}^{J}\mathrm{p}_{ij}$. Given this, note that

$$(34) \qquad \mathrm{p}_{ij} = \frac{exp(\beta_j x_i)}{(1+\sum_{k=2}^{J} exp(\beta_k x_i))}$$

and

$$(35) \qquad \log(\frac{p_{ij}}{p_{i1}}) = \beta_j \mathrm{x}_i, \qquad \mathrm{j} = 2, 3,...,\mathrm{J}$$

The parameters of this model are estimated using maximum likelihood and standard methods of inference are applicable. Details are provided in Agresti (2002) [1] and Faraway (2006) [28].

Although LG, like log-linear models, does not ordinarily distinguish between response and explanatory variables, it is, as noted above, structurally similar to the multinomial logit model and generates similar output, as well. These similarities can be exploited to gain insight into the regression model. With regard to output, suppose we have two variables, one a binary variable with m_1 categories and the other either binary or fuzzy with m_2 categories. Applying the LG algorithm to this data results in a plot containing points for the n objects along with the $m_1 + m_2$ categories. These points are positioned so that the distances from the object points to the category points reflect the probability profiles of each object according to (1), which we can think of as the LG link function. If we take the first variable to be a response variable, we can use the LG link to convert the object-to-category distances to probabilities giving us model probability output for each object of the type we obtain from multinomial regression. Also, applying the LG link to the distances between each of the m_1 response category points and the m_2 explanatory categories gives a maximum likelihood predictor profile of each response. An example of this is provided below. In some cases, standard errors and significance levels can be obtained through resampling methods.

To see the structural similarity, consider data consisting of n objects measured on some number of variables. Suppose the first is a binary indicator. The second can be either a binary or a fuzzy indicator, but suppose for now that the explanatory variable is binary or, indeed, that we have a number of binary explanatory variables. In a well-fit LG model; i.e., one with low APWL, the binary response probabilities (either ones or zeros) will be approximated almost exactly. Thus, taking objects with the same explanatory profile and averaging their response probabilities amounts to a simple counting operation. The resulting probabilities will be essentially identical to the given empirical frequencies. LG can then be used to graphically compare the model probabilities with the empirical ones. This is done by viewing the model and empirical probabilities as fuzzy indicators and constructing LG models of both.

An illustration of this idea is helpful to its understanding. Consider the data in Table 2.1. It shows a four-way classification of 1681 renters in Copenhagen who were surveyed on their type of housing, level of contact with other residents, feeling of influence on their property management, and level of satisfaction with their housing conditions. It is given in Venables & Ripley (2002) [66], having been taken in from a study originally by Madsen (1976).

We are interested in studying a main effects regression model with satisfaction as the response variable. We will use LG as described above to set a sort

Contact		Low		High			
Satisfaction	Low	Med	High	Low	Med	High	
Housing	Influence						
Tower Blocks	Low	21	21	28	14	19	37
	Medium	34	22	36	17	23	40
	High	10	11	36	3	5	23
Apartments	Low	61	23	17	78	46	43
	Medium	43	35	40	48	45	86
	High	26	18	54	15	25	62
Atrium Houses	Low	13	9	10	20	23	20
	Medium	8	8	12	10	22	24
	High	6	7	9	7	10	21
Terraced Houses	Low	18	6	7	57	23	13
	Medium	15	13	13	31	21	13
	High	7	5	11	5	6	13

Table 2.1: Copenhagen Housing Data

of baseline for comparison to the regression. First, we convert this data to four binary indicator matrices, one with two columns, two with three, and one with four, and each with 1681 rows. With three explanatory variables, we expect a three-dimensional LG model to be adequate. It gives us an APWL of .0021 with correct classification of all objects on all variables[2], which is excellent fit. Note that there are 24 explanatory profiles for objects. The objects in each profile will be divided by LG into three points, depending on their satisfaction level. As noted above, with APWL this low, when we average the satisfaction probabilities for these objects, the result is essentially identical to the empirical satisfaction probabilities. These are displayed in Table 2.2 and it can be easily checked that this is the case.

Next, we row-bind the *contact* halves of this matrix to form a 24 × 3 fuzzy indicator matrix, in which the objects are the 24 explanatory profiles. A two-dimensional uLG-1 plot of the profiles is shown in Figure 2.11. APWL is .0071 with 100% classification accuracy. It is a little crowded, but some structure is discernible. In particular, residents of Tower Blocks appear to have higher satisfaction in general while residents of Terraced Houses tend to be lower. Influence seems to be positively correlated with satisfaction as well.

We can see these relationships more clearly by plotting the groupings of explanatory profiles. The next three figures (2.12 - 2.14) do this, grouping by type of housing, level of influence and level of contact respectively. Along with the observations above, we see that apartments tend to provide for lower satisfaction and the contact seems important, but far less so than influence. Of interest to us, now, is whether the main effects multinomial regression model will show these

[2] Here, as in what follows, classification is based upon largest model category probability or, equivalently, correct Voronoi cell assignment.

Contact		Low		High			
Satisfaction	Low	Med	High	Low	Med	High	
Housing	Influence						
Tower Blocks	Low	.30	.30	.40	.20	.27	.53
	Medium	.37	.24	.39	.21	.29	.50
	High	.17	.19	.63	.10	.16	.74
Apartments	Low	.60	.23	.17	.47	.28	.26
	Medium	.36	.30	.34	.27	.25	.48
	High	.27	.18	.55	.15	.25	.60
Atrium Houses	Low	.40	.28	.31	.32	.37	.32
	Medium	.28	.29	.43	.18	.39	.43
	High	.27	.32	.41	.18	.26	.56
Terraced Houses	Low	.58	.20	.22	.61	.25	.14
	Medium	.37	.32	.32	.48	.32	.20
	High	.30	.22	.48	.21	.25	.54

Table 2.2: Copenhagen Housing Data - Empirical Probabilities (by LG)

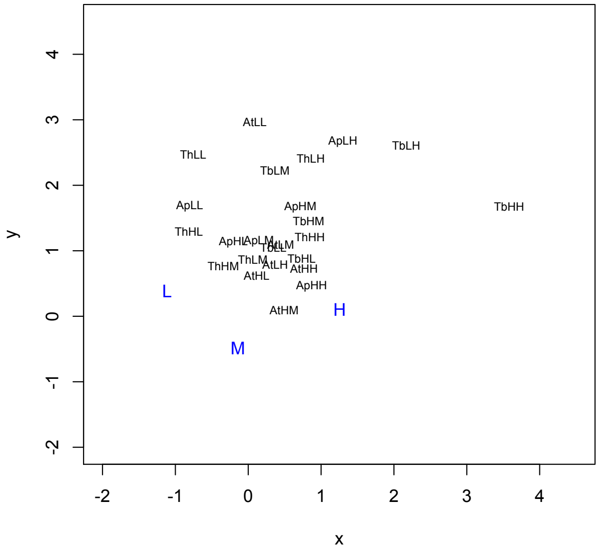

Figure 2.11: Copenhagen Housing Explanatory Profiles - LG Model

same relationships.

The regression model we will use is presented in Venables & Ripley (2002) [66]. They use a surrogate Poisson model computed with the `glm` function from the R package `stats` to fit a corresponding main effects multinomial regression. Conventional residual analysis shows it to be of satisfactory fit with no serious problems with model assumptions. They obtain the model probabilities shown in Table 2.3. Notice that these are close to the empirical probabilities, but, there are some fairly large differences.

As above, we convert these probabilities to a 24 × 3 fuzzy indicator and compute an 2D uLG-1 model. The plot is shown in Figure 2.15 on the right of the LG plot, which also appears in Figure 2.11 above. It has APWL of .0083 with 100% classification accuracy. Since the categories (i.e., satisfaction levels) are plotted similarly, the two plots can be compared fairly easily. The smoothing effect of the

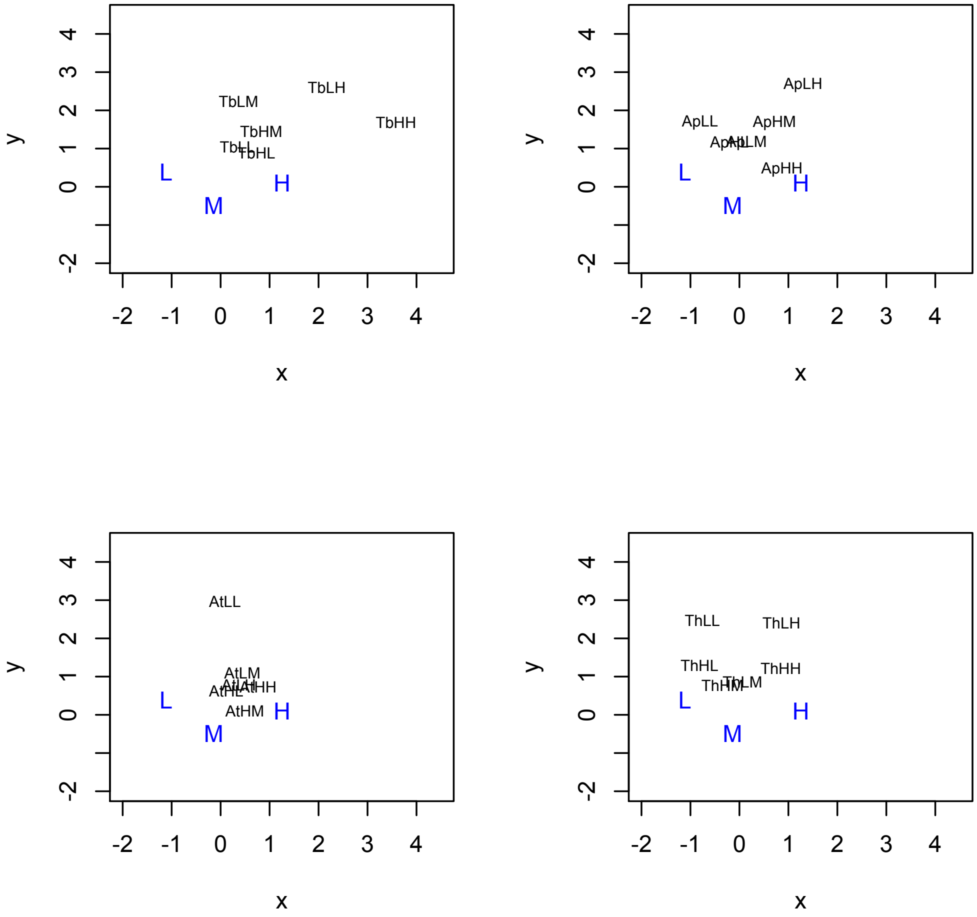

Figure 2.12: Copenhagen Housing Types - Empirical Probabilities

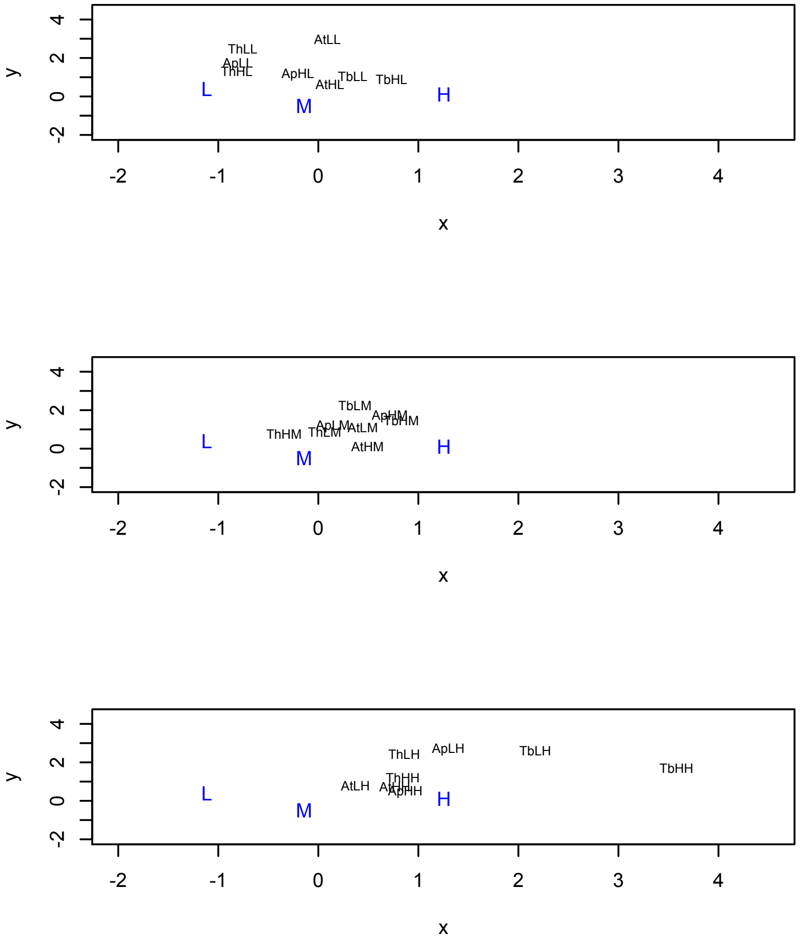

Figure 2.13: Copenhagen Housing Influence Levels - Empirical Probabilities

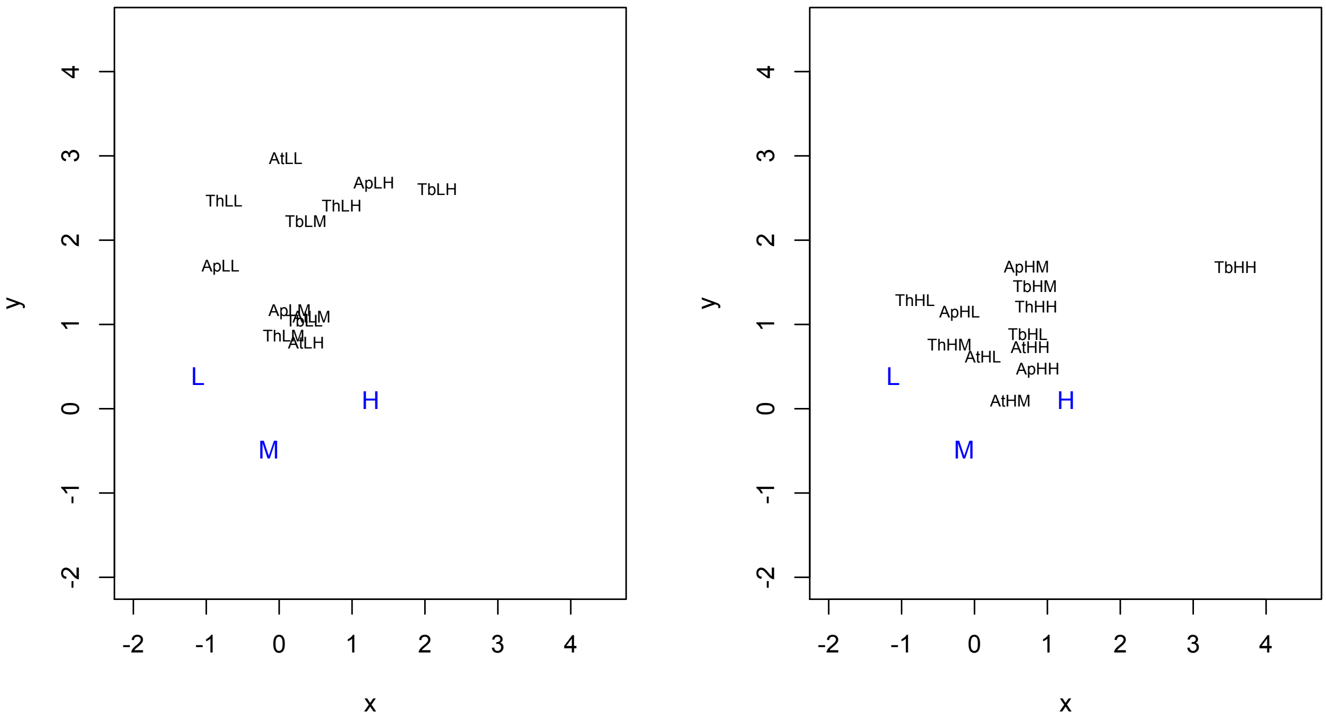

Figure 2.14: Copenhagen Housing Contact Levels - Empirical Probabilities

Contact		Low		High			
Satisfaction	Low	Med	High	Low	Med	High	
Housing	Influence						
Tower Blocks	Low	.40	.26	.34	.30	.28	.42
	Medium	..26	.27	.47	.18	.27	.54
	High	.15	.19	.66	.10	.19	.71
Apartments	Low	.54	.23	.23	.44	.27	.30
	Medium	.39	.26	.34	.30	.28	.42
	High	.26	.21	.53	.18	.21	.61
Atrium Houses	Low	.43	.32	.25	.33	.36	.31
	Medium	.30	.35	.36	.22	.36	.42
	High	.19	.27	.54	.13	.27	.60
Terraced Houses	Low	.65	.22	.14	.55	.27	.19
	Medium	.51	.27	.22	.40	.31	.29
	High	.37	.24	.39	.27	.26	.47

Table 2.3: Copenhagen Housing Data - Regression Model Probabilities

regression is evident in its plot, and some of the relationships we observed above are discernible, but not quite as clearly as before.

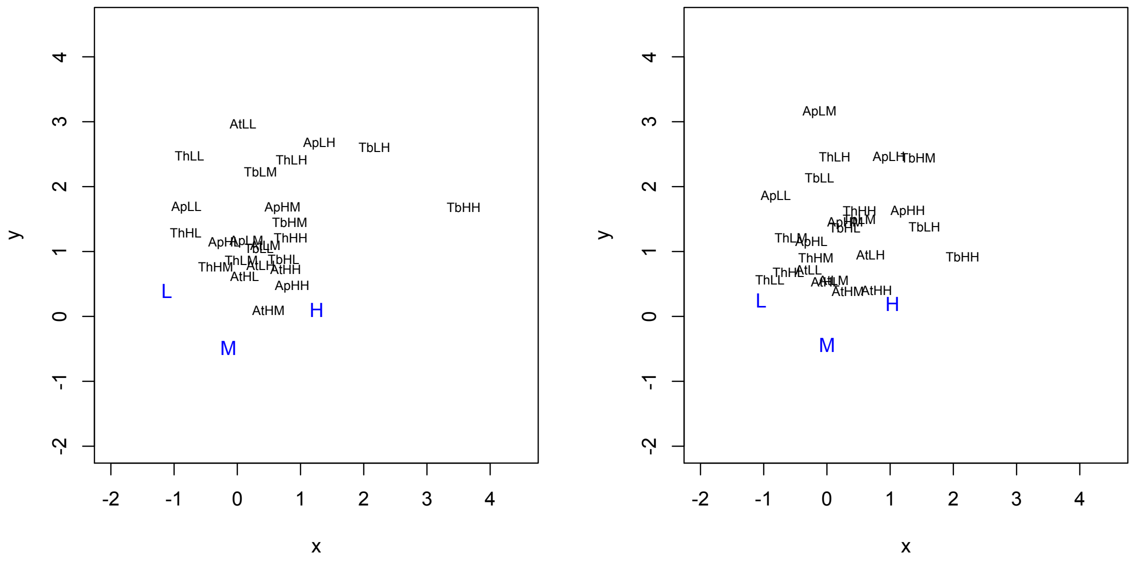

Figure 2.15: Copenhagen Housing Profiles- Empirical and Regression Models

As before, it is useful to plot the profile groupings separately. In Figure 2.16, we see the housing grouping. Comparing it to Figure 2.12 above, we see that the regression smoothes somewhat the effect of the Terraced Housing variable. Also, the regression views the Atrium Houses as slightly favoring medium and high satisfaction, whereas the empirical probabilities are more neutral. In Figure 2.17, the influence groupings are shown together with Figure 2.13, the influence groupings of the empirical data, here with the regression on the bottom row. In both models, the relationship between influence and satisfaction is displayed, though again, we notice in the regression some smoothing of the Tower Blocks effect. Finally, the contact groupings are displayed together, again with the regression on the bottom. We see that the contact effect, in itself, is slightly larger in the data than in the regression model. Overall, we have clearly displayed that the regression model fits the data quite well.

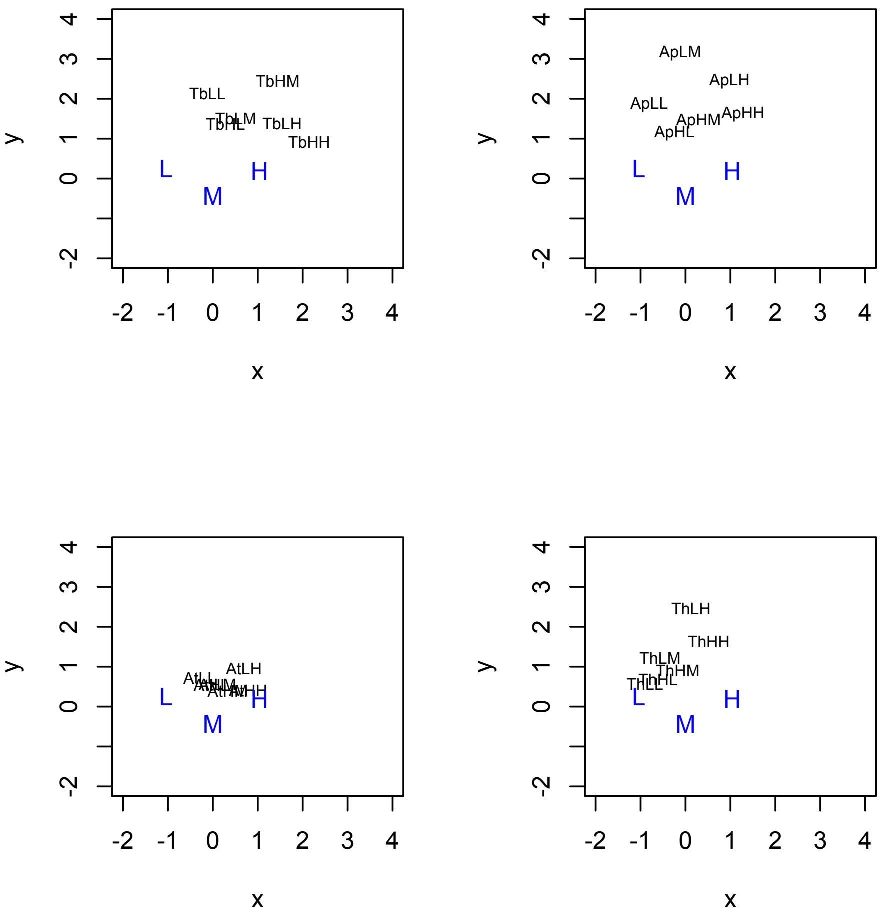

Figure 2.16: Copenhagen Housing Types - Regression Model

Using constrained LG[3] we can extend the above approach to provide a visualization of multinomial regression model testing. This approach is particularly useful in situations where we have some extrinsic or a priori knowledge or beliefs about the objects involved. An LG plot and a multinomial regression can be com-

[3] As we will discuss below, constrained LG involves unfolding with one set of coordinates (usually category points in our work) fixed for some theoretical reason. In MDU literature, it is sometimes referred to as *external* unfolding.

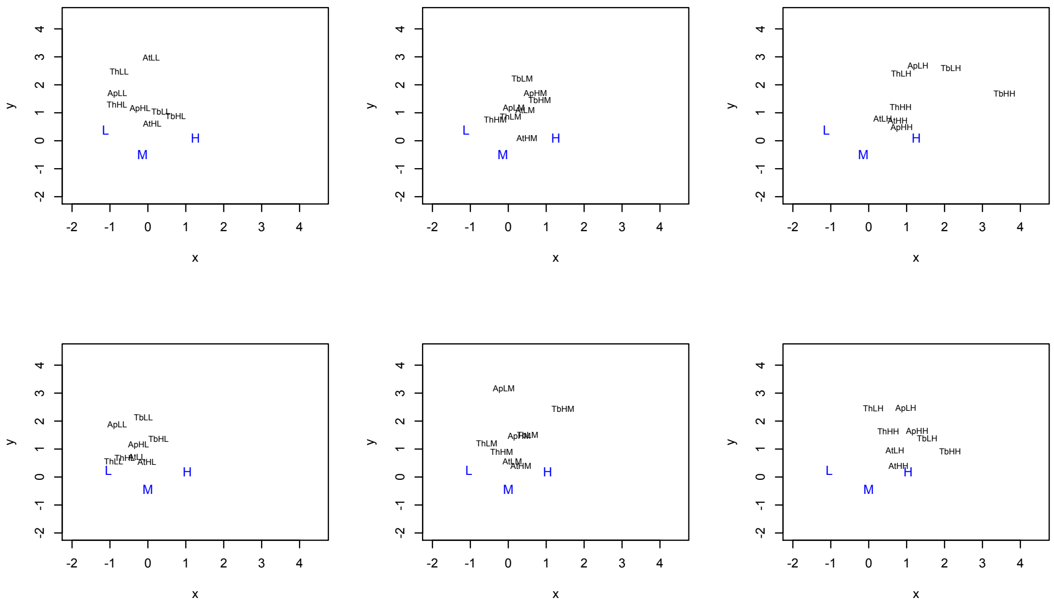

Figure 2.17: Copenhagen Housing Influence Levels- Empirical and Regression Models

puted using a training sample. The LG plot provides a configuration of category points (the *training configuration*) and the regression a set of coefficients from which classification predictions can be made. With the training configuration fixed, the LG model can be run on the testing data to see how well coordinates can be computed with the category constraints. Then, regression predictions can be made using the testing sample and these can also be modeled using the constrained LG algorithm. The resulting plots can be compared to assess graphically the fit of the linear model, somewhat akin to an actual-vs.-predicted plot, which is otherwise difficult to construct for multinomial regression.

To illustrate this idea, we will examine a dataset taken from the Mapping LA Neighborhoods data maintained by the LA Times. (See the supplementary file.) There are 2 variables, hence two indicators. The first is a compositional variable giving the percentage of Asians, Blacks, Latinos, and Whites in each neighbor-

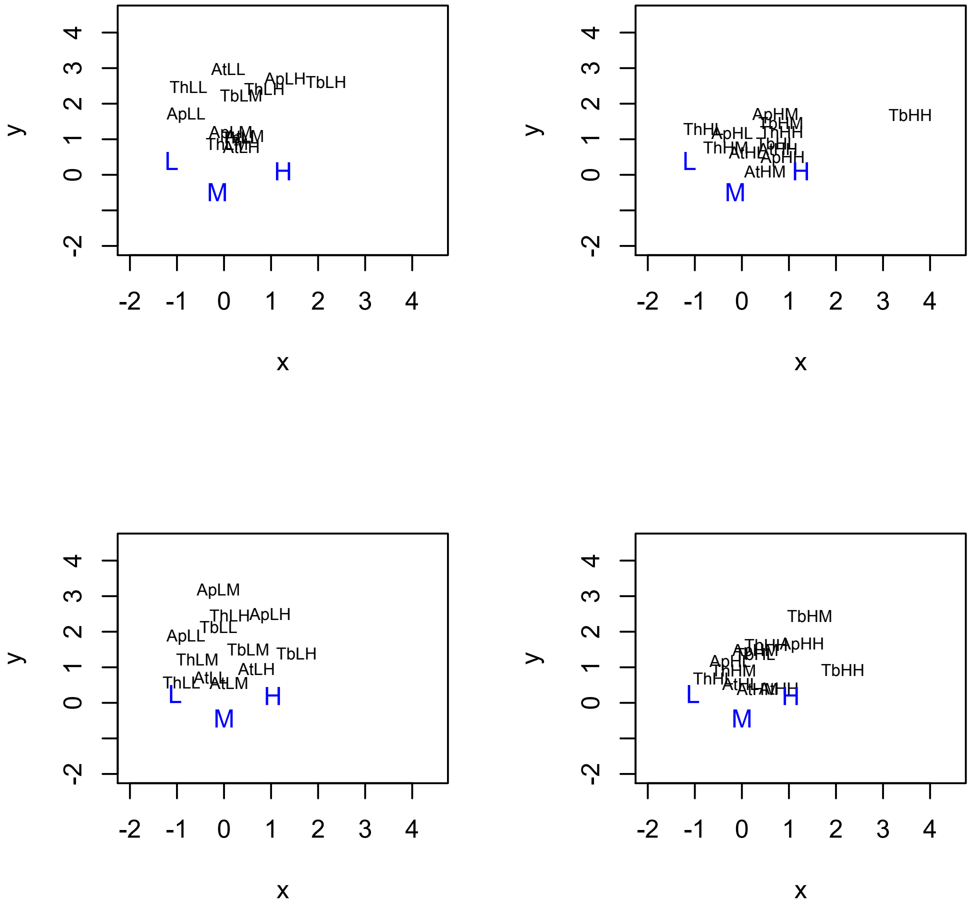

Figure 2.18: Copenhagen Housing Contact Levels- Empirical and Regression Models

hood, producing a fuzzy indicator, and the second an ordinal variable classifying the median income of each neighborhood as Lower, Lower-Middle, Upper-Middle, or Upper, producing a 4-column binary indicator. Fifty-three neighborhoods were selected at random from the dataset. For 8 of these, the four ethnic groups above did not account for 100% of the ethnic makeup, with other ethnicities accounting for from 1% to 4%. For these 8 objects, the four main ethnic group percentages were proportionately scaled to total 100%. Since our goal here is mainly illustrative, this very small modification of the data can be accepted to avoid working with a category of mainly 0's and a few very small non-zeroes.

Our uLG-1 algorithm was run for two dimensional output with maximum iterations at 1500, experience showing this to be an appropriate point to end the computation for this dataset. Figure 2.19 shows the resulting plot (with some jittering of object points to enhance legibility). The APWL for the fuzzy indicator portion of the data is .03919 and for the binary portion .00631. These results are quite good for mixed indicator data. Further, all 53 neighborhoods are correctly classified by income quartile. We generally prefer having the clarity of a 2D data visualization and are able to have that here since the APWL improvements seen in higher dimensions can generally be considered marginal.

The plot has some interesting features, showing the correlation structure between neighborhood ethnicity and income and identifying a handful of distinctive neighborhoods. It will be useful, here, to examine it in some detail keeping in mind our principles of LG interpretation. First, note that the neighborhoods are clustered by the binary variable, income quartile, which efficiently limits a source of APWL. Next, we see that the Asian and White ethnicities are fairly closely associated with the Upper Middle quartile, with White having considerably higher association with the Upper quartile. (Remember, the small shift of White toward the Upper category point is quite significant in LG.) Black and Latino are more

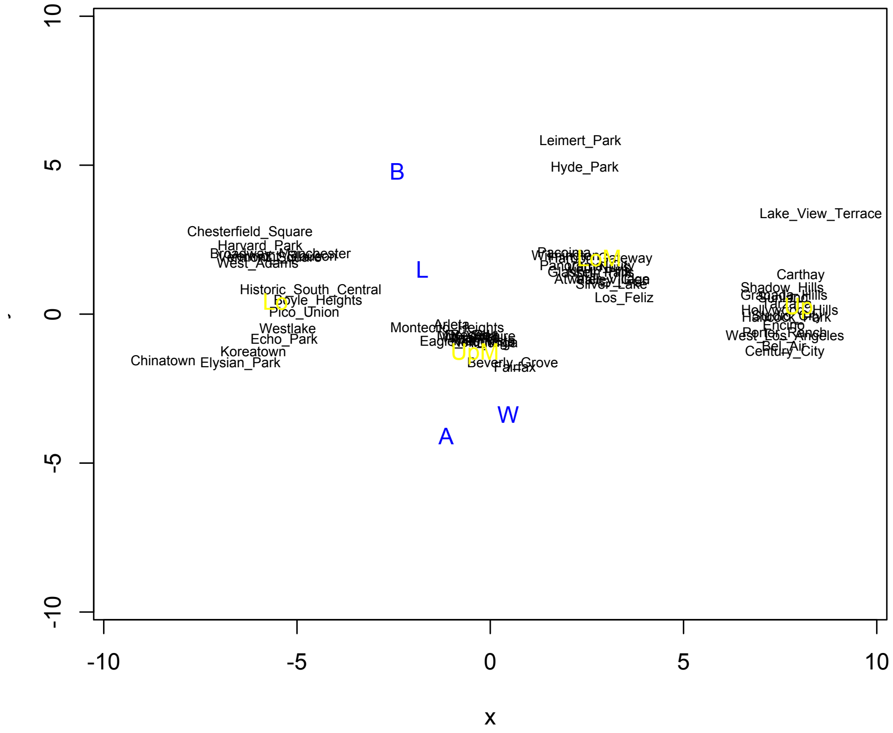

Figure 2.19: LA Neighborhoods

closely associated with the Low and Low Middle quartiles, with Latino showing much greater association than Black with Upper Middle. The slight pulling-apart of the clusters is of interest, showing that Low quartile neighborhoods have some diversity of mixture among Asian, Black, and Latino, while, among the other three quartiles there appears to be considerably less ethnic diversity, with only 3 neighborhoods having somewhat higher Black and Latino percentages. Table 2.4 shows the explanatory variable profiles under this model for the income quartiles. It clearly shows that the Upper and Upper Middle income areas are predominantly White and does appear to give some evidence supporting greater movement to upper income areas within the city among Latinos and Asians than among Blacks.

	Asian	Black	Latino	White
Low	0.080	0.135	0.766	0.019
LoM	0.048	0.175	0.601	0.176
UpM	0.194	0.013	0.264	0.530
Up	0.113	0.018	0.119	0.751

Table 2.4: Income Quartile Ethnic Profiles for uLG Plot

It is of interest to compare Table 2.4 with the mean ethnic group percentages among income quartiles in the data. Table 2.5 gives those and we can see that, though there are some relatively large cell differences, the LG profiles of Table 2.4 capture very well, and emphasize, the patterns in the data noted above, which are also reflected in Table 2.5. It should be kept in mind that the profiles in both tables are to be taken as profiles of typical objects in the dataset; they do not give an estimate of the overall ethnic makeups of persons at these income levels in the population at large.

Next, we fit a multinomial regression to this data using income quartile as the response. This is an ordinal response variable so that an adjacent-categories

	Asian	Black	Latino	White
Low	0.118	0.223	0.621	0.038
LoM	0.084	0.185	0.543	0.187
UpM	0.133	0.052	0.380	0.435
Up	0.106	0.036	0.121	0.738

Table 2.5: Income Quartile Mean Ethnic Profiles

or cumulative logit model may be most appropriate; however, we shall use the multinomial logit model since it is closest in structure to the uLG-1 link which is our main interest. Some loss of statistical significance in the regression results is, therefore, to be expected, but is not of great concern at this descriptive stage of analysis.

The model is fit using the function `multinom` from the R package, `nnet` of Venables & Ripley (2002) [66]. It uses neural network methods to compute maximum likelihood parameter estimates. Table 2.6 shows the model summary. Note that fairly high significance levels are attained for the Black, Latino, and White coefficients. We are interested in using a testing sample to evaluate this model against the uLG-1 model and in using uLG-1 to visualize the testing performances.

A random sample of 12 LA neighborhoods was selected from the Mapping LA Neighborhoods data. We first attempted to fit the data to the LG model by inputting it into a version of the uLG-1 algorithm which fixes the category points in Figure 2.19 (and which we will refer to as *fixed category* or *constrained* LG). Model fit is comparable to the overall model, with APWL = .0462 for the ethnicity fuzzy indicator matrix and .0069 for the income quartile binary indicator. Here also, all neighborhoods are correctly classified by income quartile.

Next, we test the regression model with the same testing sample. The regres-

Coefficients:					
	(Intercept)	Asian	Black	Latino	White
2	11.46361	-71.87422	-10.47171	-17.77053	111.5801
3	11.46207	-67.52283	-15.19345	-20.60330	114.7816
4	10.90360	-71.26630	-11.48956	-24.36326	118.0227
Std. Errors:					
	(Intercept)	Asian	Black	Latino	White
2	5.791940	45.61703	5.092904	9.037179	63.39479
3	5.932054	45.84592	6.686741	9.487593	63.41959
4	5.926362	46.08585	6.244399	9.798397	63.43789

Table 2.6: Multinomial Regression Summary

sion prediction itself has an APWL of .2111 against the binary income quartile indicator, which, with a binary indicator, still allows for correct classification by the regression of 9 of the 12 neighborhoods. (This is somewhat better than the regression model's performance on the training data in which 33 out of 53 neighborhoods were correctly classified.) A fixed- category uLG-1 model was fit using the regression predictions as the indicator (a fuzzy indicator) for income quartile. Figure 2.20 shows the fixed category plot of both the uLG-1 (black) and regression (purple) test results. APWL for the regression income quartile indicator is .0424 and for the ethnicity indicator, .0829. Note that the APWL for the regression predictions is somewhat higher than the fixed category APWL's. This suggests that there is some degree of stability in the category configuration. Also, we see at a glance that the regression misclassifies East Hollywood as Low Middle, close to Upper Middle, instead of Low. East Hollywood has an 18.7% white population which is somewhat higher than usual for a Low income neighborhood. This likely accounts for the misclassification. Notice that it is not misclassified by LG, which suggests that influence points are less likely (perhaps, far less likely) in

LG than in regression. The other misclassifications are also revealing. Baldwin Hills/Crenshaw (71% black) is misclassified as Low Middle, not Low, and Pico Robertson (74% white) as Upper, not Upper Middle. Finally, an interesting feature of the plot is that the smoothing, or approximating, affect of the regression is apparent in the positioning of the regression neighborhood points along a line that seems to almost simultaneously bisect all pairs of categories, hewing closely to the Low Middle - Upper Middle bisector.

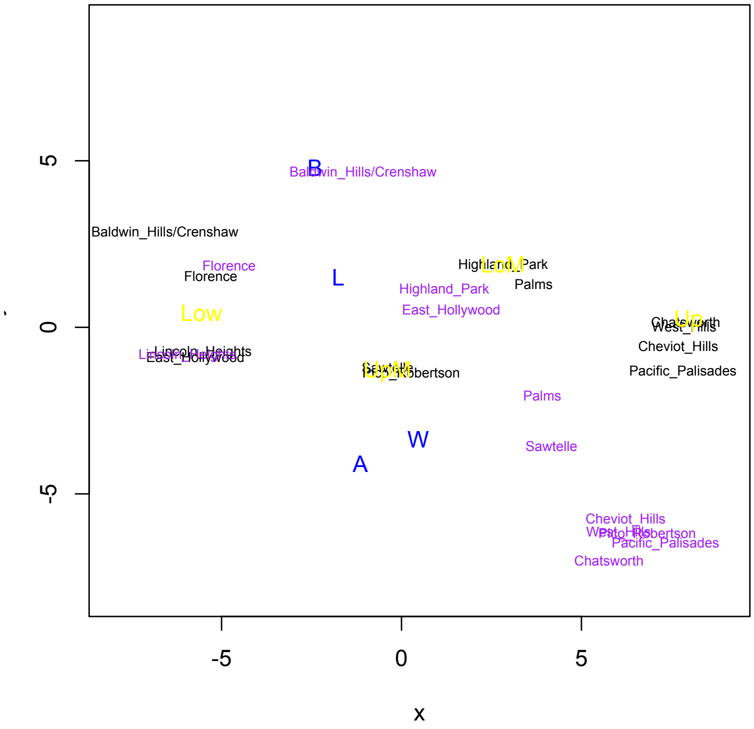

Figure 2.20: LA Neighborhoods - Actual vs. Predicted

The significance of this last observation is made clearer by examining the mathematical relationship between the multinomial logit and the LG link. Suppose that we have computed an LG model. From (29) and (35), we have:

(36) $\beta_j \mathrm{x}_i = \mathrm{d}_{i1} - \mathrm{d}_{ij}$

where the x_i's are the predictor values and the right hand side distances can be calculated from the LG model. Thus, finding the regression parameters, βj, amounts to solving an elaborate system of linear equations. This, of course, cannot be solved exactly, so a best-fitting solution is sought. As we see from our example above, some loss and some smoothing are to be expected.

2.4 Social Network Plotting

One of the most important uses of data visualization is social network plotting. Social network data can be viewed as a set of actors, or nodes (possibly with some attribute variables), and the ties between them. The ties may be non-directed or directed, and may be weighted. (Here, for the time being, we shall consider non-weighted networks.) Aspects of the network such as density, prestige (or degree) of actors, cliques of actors and brokerage among cliques, transitivity and reciprocity (for directed networks) of ties, types of stars, triads, and other formations, etc. are of interest to the researcher. An excellent comprehensive text on descriptive social network analysis (SNA) discussing these topics is Wasserman & Faust (1994) [67].

Typically, the network ties are represented by a sociomatrix, which is essentially an adjacency matrix, to borrow a term from graph theory, and the network can be visualized as the corresponding graph, digraph, or weighted graph. Of course, there are many arbitrary ways to draw the graph of any network; the goal of social network plotting is to draw it in such a way that it can assist in the description and investigation of the network by displaying features of it that may warrant further study. Our purpose here is to describe how uLG-1 can be used to construct network plots, to provide examples of such plots for three classic networks, and to compare these to plots constructed by two other popularly used algorithms.

One of these oft-used methods is known as force-based or force-directed graphing. This is the default method, for example, in the R packages `network` and `sna`. There are several different force-directed algorithms. They share a central idea which is to view the network as a physical system (of springs or electric or gravitational forces, for example) and the plotting problem as involving positioning nodes and ties in such a way that the system is placed in equilibrium. Classic expositions of the method are Fruchterman & Reingold (1991) [30] and Kamada & Kawai (1989) [43] and a fairly comprehensive survey can be found in Kaufmann & Wagner (2001) [44]. [4] The main visual characteristics of force-directed graphs are relatively uniform edge (or tie) lengths, few crossing edges, uniform node distribution and central positioning of high-degree (or highly-connected) nodes.

Pictured below are force-directed plots made in R using `network` of three classic, well-studied social networks. The first (Figure 2.21) is known as Sampson's Monks and is based on directed (i.e., possibly non-symmetric) friendship ties observed within a small monastery (Sampson (1969)). Data for this network is found in `network` and in `sna`. The common features of force-directed graphing are present; in particular, relatively even spacing among nodes and central positioning of high-degree nodes (e.g., John and Bonaventure). What does not stand out here is the clustering of the monks into three groups which was of importance in Sampson's research into this network. We will discuss this further below.

The second network plotted (Figure 2.22) is known as the Florentine Families (originally from Padgett & Ansell (1993) [51] and obtained, again, from `network`

[4] Interestingly, the spring model can be viewed as finding ideal spring lengths, then minimizing the difference between these and a set of Euclidean distances; i.e., as a metric multidimensional scaling problem. See Kaufmann & Wagner (2001) [44] on this topic.

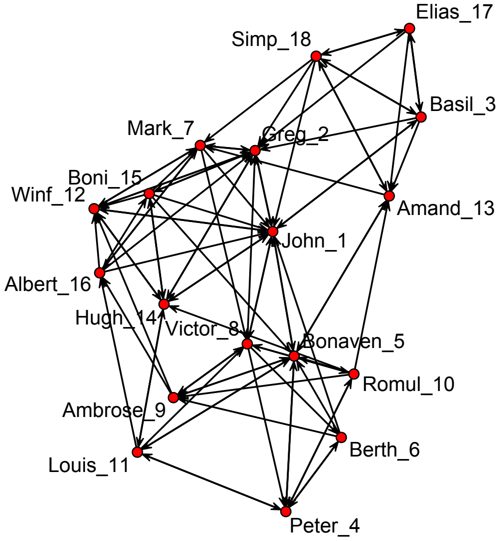

Figure 2.21: Sampson's Monks - Force Directed

and sna) and shows marriage ties among several prominent families in Renaissance Florence. It has a low density, so is very cleanly visualized by the force-directed method. Of importance here are the prominence of the Medici's, the Strozzi clique, and the relations between the two. These are displayed here, but, it is fair to say, not emphasized.

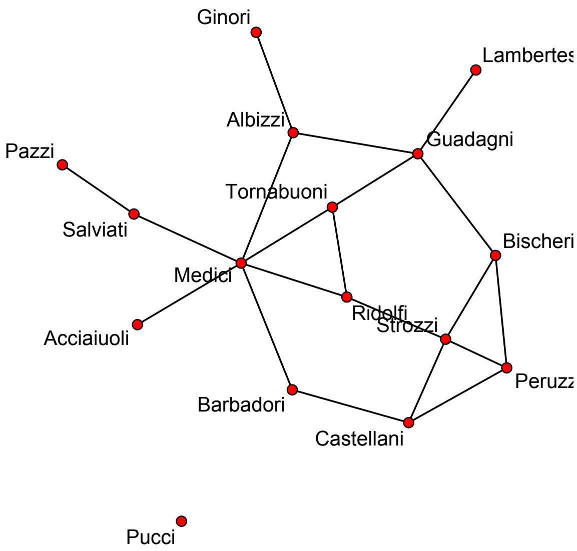

Figure 2.22: Florentine Families - Force Directed

The third network (Figure 2.23) is of friendship ties among the French Financial Elite (from network featured in a classic study by Kadushin (1995) [42]). Its most noteworthy feature, discussed at length by Kadushin, is that it appears to be the joining of two different social structures, or moieties, one of which is clearly denser than the other. Kadushin found there to be 13 actors in the less-dense moiety and found moiety membership to be related to attendance of a prestigious French business school, the ENA. Within the denser moiety (of ENA alums),

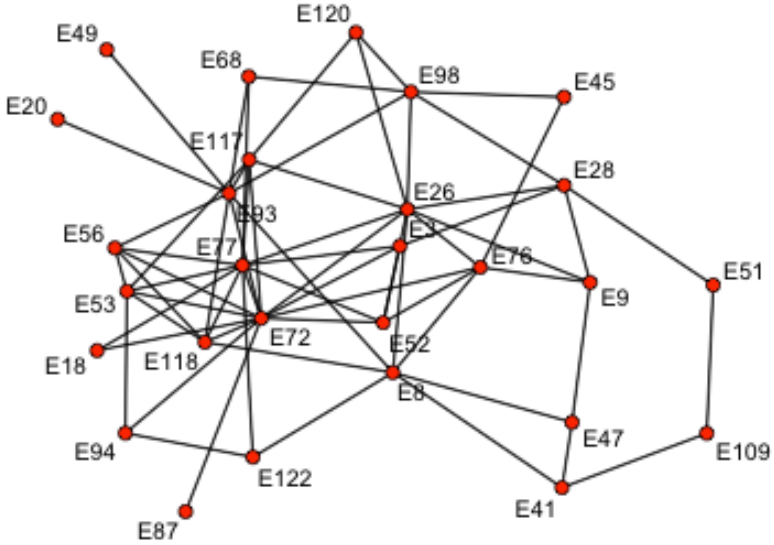

Figure 2.23: French Financial Elite - Force Directed

nodes E72 and E77 are of high-degree and centrally positioned. Nodes E8, E26, E76, and E98 play something of a brokerage role between the groups. The group of densely connected nodes is well-displayed in the plot and the moiety structure is detectable, but, again, because of the uniformity of positioning, it is not emphasized.

A second very important method of social network plotting is the latent space approach developed by Hoff, Raftery, & Handcock (2002) [41]. In the latent space models (LSM), nodes are positioned so that the probability of a tie between any two is given by a function of the distance between them in a latent social space. At the conceptual level, the approach has much in common with LG. In fact, a logistic regression model is used to parameterize the likelihood of the network configuration. Letting y_{ij} be a binary variable indicating presence or absence of a tie between nodes i and j, the model is given by:

$$\text{(37)} \qquad \log \text{odds}(\mathrm{y}_{ij} = 1|\mathrm{z}_i,\, \mathrm{z}_j,\, \mathrm{x}_{ij},\, \alpha,\, \beta) = \alpha + \beta \mathrm{x}_{ij} \text{ - } \|\mathrm{z}_i \text{ - } \mathrm{z}_j\|$$

where α is a network density parameter, x_{ij} are observed dyad covariates, β is a vector of coefficients, and the z_i's are node position coordinates. Under the assumption that the value of y_{ij} is independent of all other ties in the network given the positions of the two nodes, the parameters and distances between nodes are estimated using maximum likelihood methods. The position coordinates are then computed from the distances using MDS methods, with MCMC methods used to determine confidence regions for the positions. The function `ergmm` in the R package `latentnet` computes the positions under various modeling options. Note that in Euclidean space, LSM favors reciprocal and transitive tie structures because of these properties of Euclidean distances. Also, in both this model and the Biased LG model (introduced in footnote 3 of Chapter 1 and to be discussed below), the log odds ratio of a tie from node i to node j versus i to k is the product of a ratio of bias or adjustment parameters and the differences in the distances from i to j and i to k.

Pictured in Figures 2.24 - 2.26 are LSM plots for the same three classic networks shown above produced by `latentnet`. For Sampson's Monks, this plot has clearly separated the monks into 3 clusters. Notice that John and Bonaventure are no longer classed together due to their centrality, but, instead, are easily seen as belonging to separate clusters. These clusters correspond to Sampson's identification of three factions in the monastery (named, *young Turks*, *loyal opposition*, and *outcasts*), thus the plot is a useful visualization of this aspect of Sampson's results. Next, is the LSM plot of the Florentine Families. It is structurally similar to the force-directed plot, clearly emphasizing the prominence of the Medicis through central positioning. Also, it more clearly highlights the Strozzi clique. However, the plot may be considered slightly misleading since two other families appear nearly as central as the Medicis. Figure 2.26 is the `latentnet` LSM of the French Financial Elite. It is, frankly, difficult to discern Kadushin's moiety structure in the plot as opposed to a central-versus-peripheral structure (which, we should say, may itself be worthy of study). Figure 2.27 is the same network, this time plotted with functional parameters specifically set to find 2 groups in the network. The moiety structure is now somewhat more apparent, with `latentnet` having identified (by yellow vertices) 7 of the 13 members in the less-connected group.

We next discuss network plotting using uLG-1. To represent the network in the proper form for a standard application of LG (i.e., as a set of indicator matrices each giving category membership for a collection of objects), we consider the network nodes to be both objects and variables. This is similar to what is known as *dédoublement* in the French correspondence analysis school. Thus, if there are n nodes, there are also n variables and n indicator matrices each of

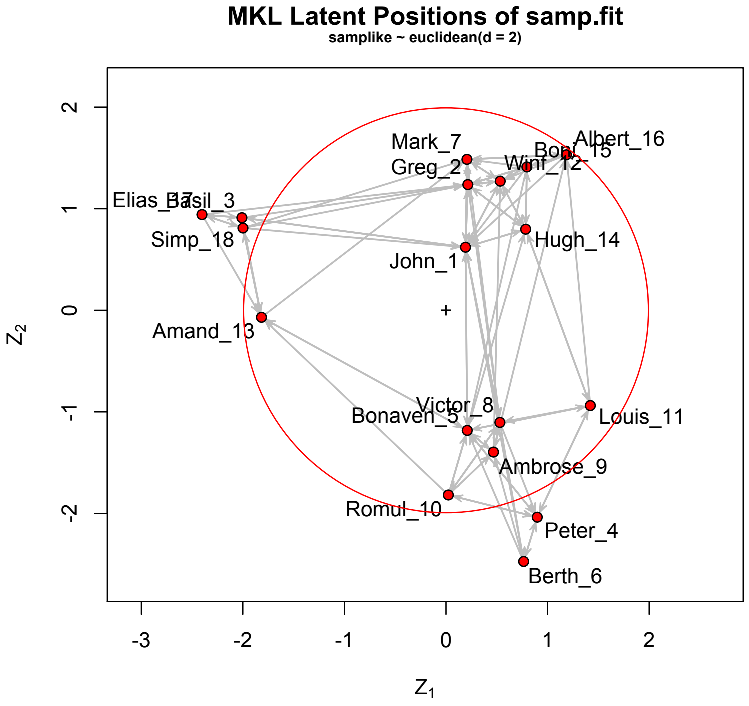

Figure 2.24: Sampson's Monks - Latent Space Model

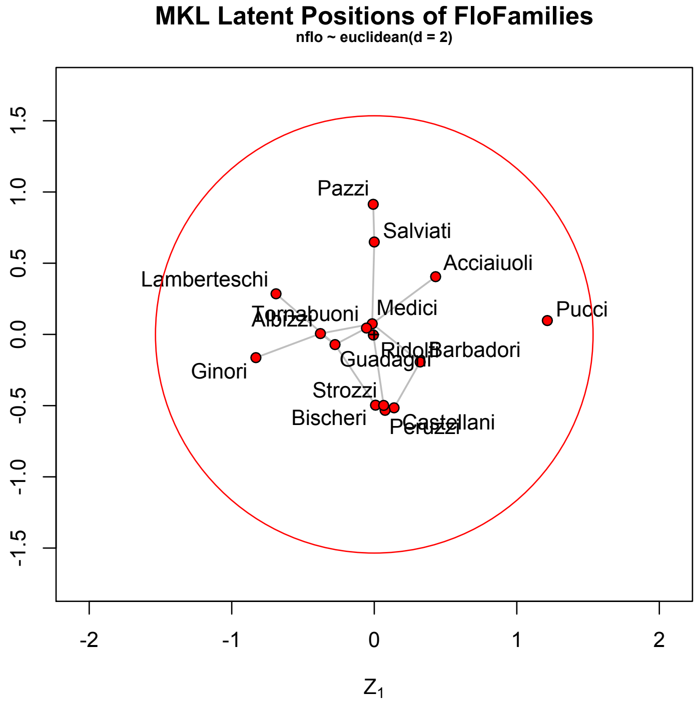

Figure 2.25: Florentine Families - Latent Space Model

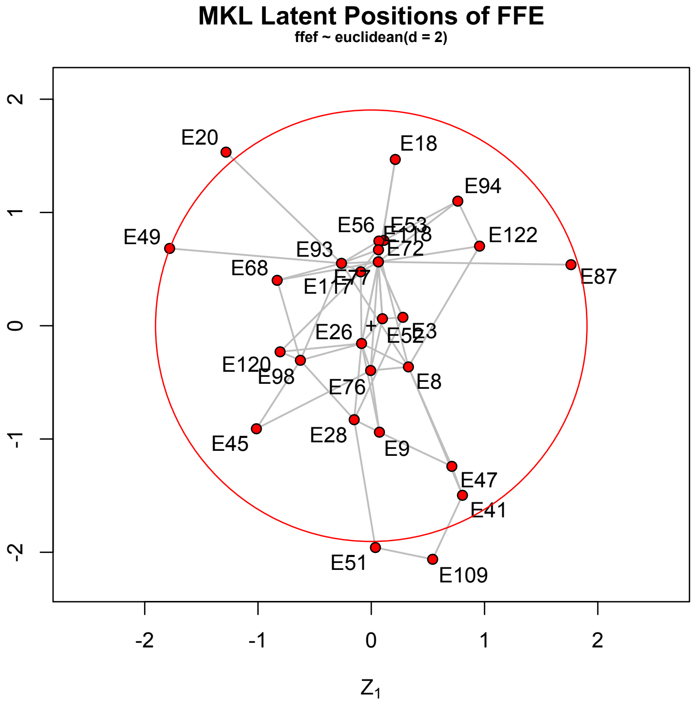

Figure 2.26: French Financial Elite - Latent Space Model

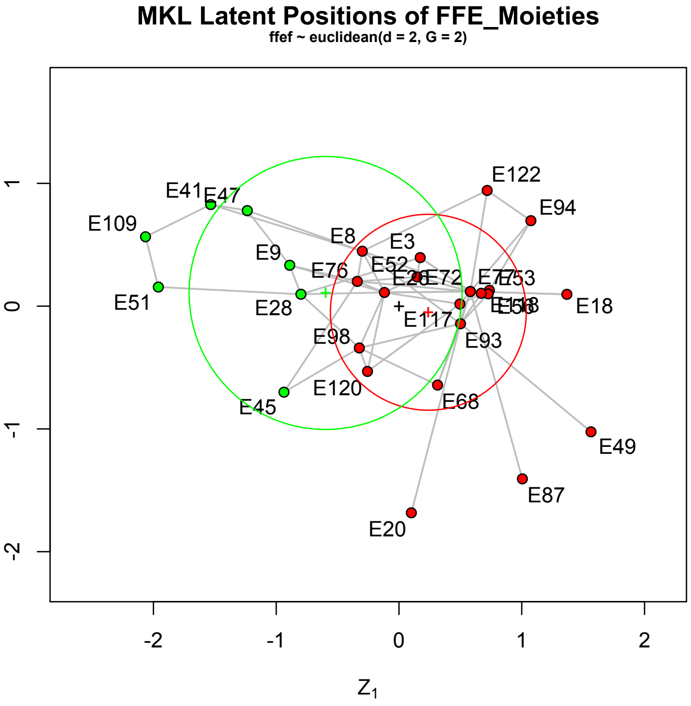

Figure 2.27: French Financial Elite - Two-Group Latent Space Model

dimension $n \times 2$; one column for absence of a tie, the other for presence. Since, ultimately, LG classifies by object profile, we consider each node to have a tie with itself; i.e., each node belongs to its own tie category so that it will be seen as similar to other nodes with ties to it.

The LG plot of Sampson's Monks (Figure 2.28) is strikingly similar to the LSM plot, clearly showing the same clustering pattern. This clustering, then, can be seen to be based upon tie profile independently of any node attributes.

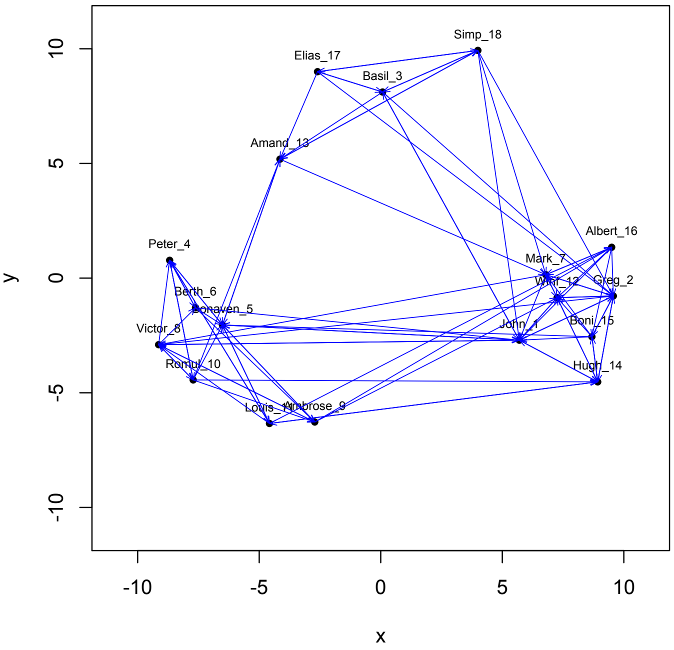

Figure 2.28: Sampson's Monks - LG Model

On the other hand, the LG plot of the Florentine Families (Figure 2.29) is quite different from its counterparts. Rather than placing the Medicis at the center of the network, it has plotted the network hierarchically with the Medicis at the apex. This goes slightly against the grain of traditional network plotting, but it is difficult to say that the LG plot does not emphasize the prominence of the Medicis, perhaps better than both the force-directed and LSM plots. Also, in the LG plot, the Strozzi clique is at essentially the opposite position from the Medicis, giving another useful visualization of an important feature of the network.

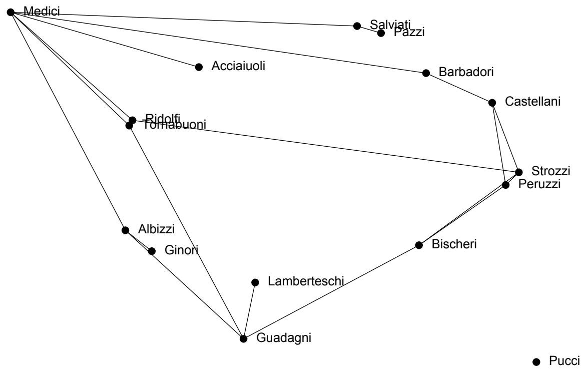

Figure 2.29: Florentine Families - 2D LG Model

A 3-dimensional LG plot of the Florentine Families greatly improved APWL (.09998 vs. .01148). It is shown in Figure 2.30. Again, it is hierarchical in structure with the Medicis at its apex. Notice that it is the network structure produced by the Strozzi (plotted by a lower-case s) clique that seems to require the 3-D configuration for accurate visualization and it is likely that this is the source of the greatly improved APWL. (Note that due to labeling limitations, the Pucci family, the network isolate, is marked with an X.)

A brief discussion of the construction of the Florentine Family plots is of value

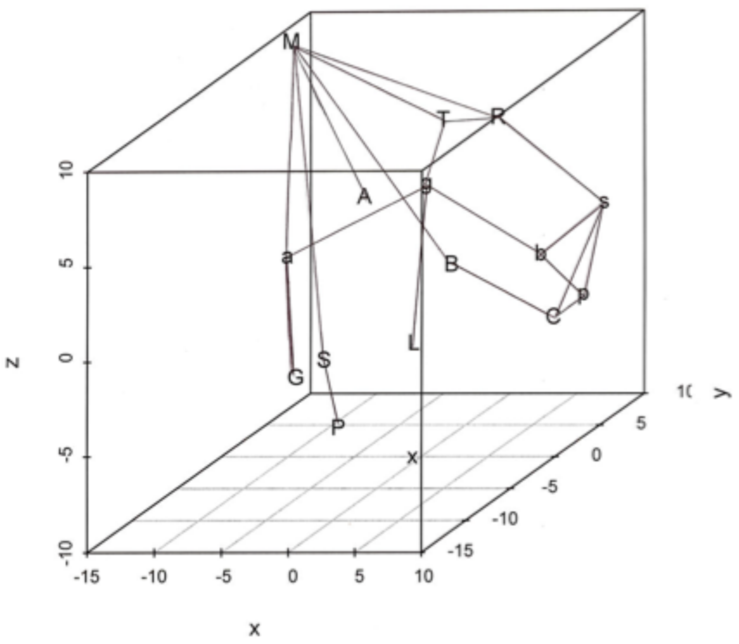

Figure 2.30: Florentine Families - 3D LG Model

in understanding the network plotting of LG and, in particular, the hierarchical structure of this plot and the one to follow. In the LG scheme using dédoublement, each node is represented by 3 points: its object point and two category points, one category point for a tie (or connection) and one for no tie. The LG algorithm (uLG-1 in these examples) finds optimum positions for all 3 of these points though our network plots show only object points since they are the most readily interpretable. Each object will be fairly closely associated with its tie category point and the other tie categories it is in and relatively far from the no-tie categories it is in. For a low-density network like the Florentines with an isolate node (the Puccis), all object points must be relatively close to the Pucci no-tie point and each object must be relatively close to several other no-tie points. To efficiently reduce APWL, the algorithm centrally places these no-tie points and moves object points and tie category points in an approximately concentric pattern around them. This produces the hierarchical appearance of this plot as well as of the French Financial Elite, Figure 2.31.

Given the nature of the French Financial network, this does not seem inappropriate. Actors E72 and E77 are most prominent and are at the apex of the network. Their very high prestige, as reflected by their network tie degrees, is very clearly displayed. Also, the plot effectively visualizes the moiety structure of the network. Eleven of the 13 members of the lesser moiety are easily spotted in the plot. Being an ENA alum is thought to be a significant classifier of moiety membership, but this plot shows another possible distinguishing characteristic: direct ties with E72 and E77.

With ENA in mind, an indicator for ENA attendance was added to the data and a new uLG-1 plot was produced, shown in Figure 2.32. The ENA attribute produces a two-category indicator so this is something akin to instructing `latentnet` to look for two groups in a latent space plot as was done above. ENA

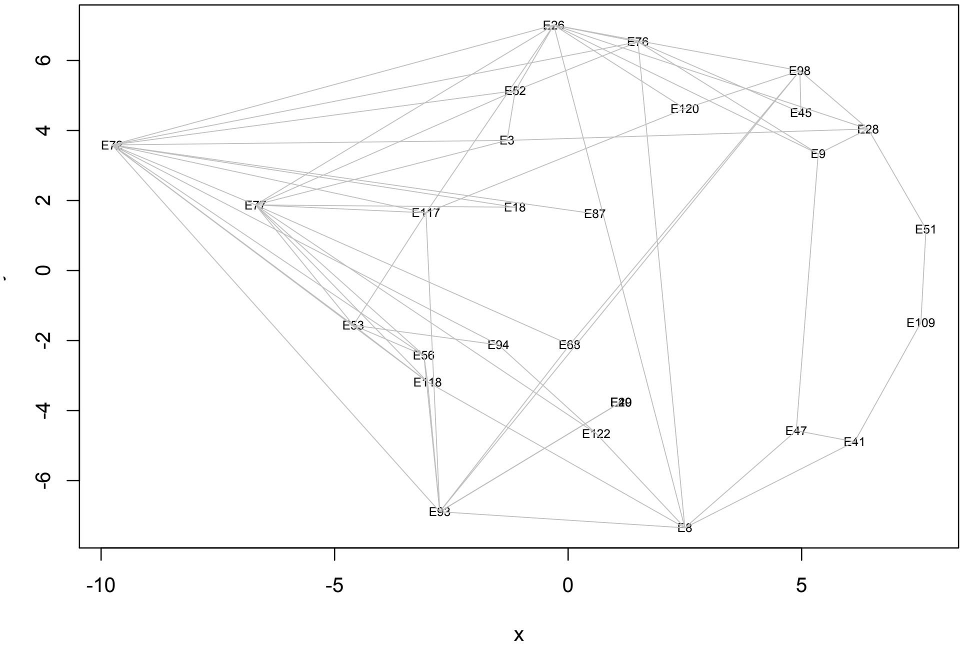

Figure 2.31: French Financial Elite - LG Model

membership does appear to be associated with moiety membership and the plot does not appear to be much different. APWL changes only very slightly, from .1343 to .1377. Closer examination, however, reveals that moiety membership is now easily discernible only for 9 of the 13 members of the less prestigious group. These are the 9 non-ENA members on the NoENA side of the gray dashed line which bisects the ENA-NoENA segment. The four ENA members of the lesser group are now shifted toward the more prestigious group (with only 2 non-ENA alums out of 15) or, more accurately, toward the ENA category. Thus, accounting for ENA attendance disturbs slightly the moiety structure. It would likely be of interest to the researcher to test other node attributes (such as political party) or combinations thereof in the same way and to compare the results with those for ENA. These observations, as well as those made about the other networks and network plots, suggest that LG is a promising method for exploratory SNA.

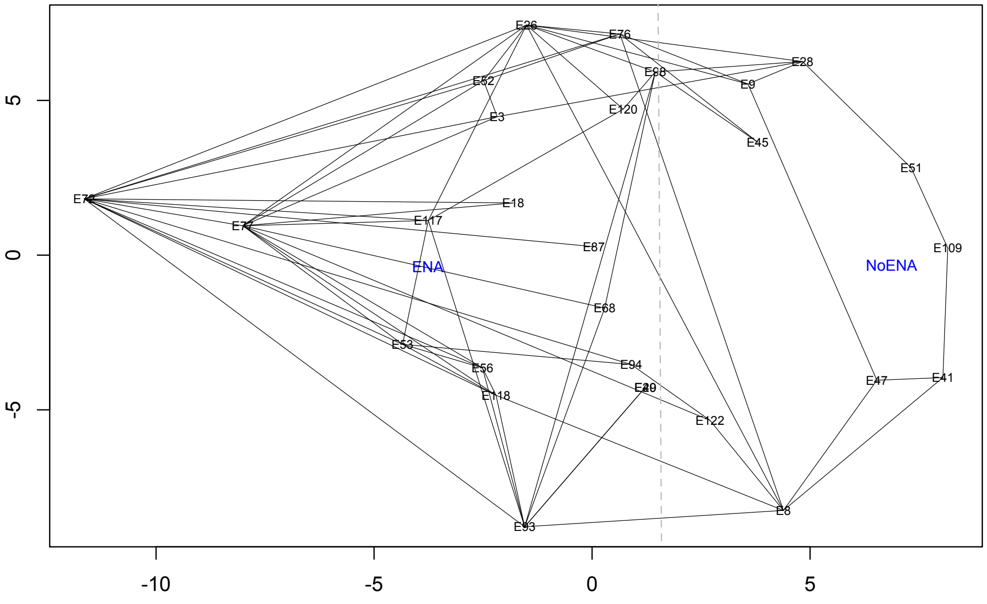

Figure 2.32: French Financial Elite w/ ENA - LG Model

Through the use of fuzzy indicators, LG can be used, as well, in connection with probabilistic plotting of networks to give a more conventional view of network structure. This method also provides a straightforward way to plot weighted networks. We will simply add 1 to the diagonal of the adjcency matrix then divide each row by its row sum to create a fuzzy indicator, then plot this with LG. Our first example is shown in Figure 2.33 (with a magnification in Figure 2.34) for the French Financial Elite. The plot appears drastically different from the dédoublement-style LG plot.

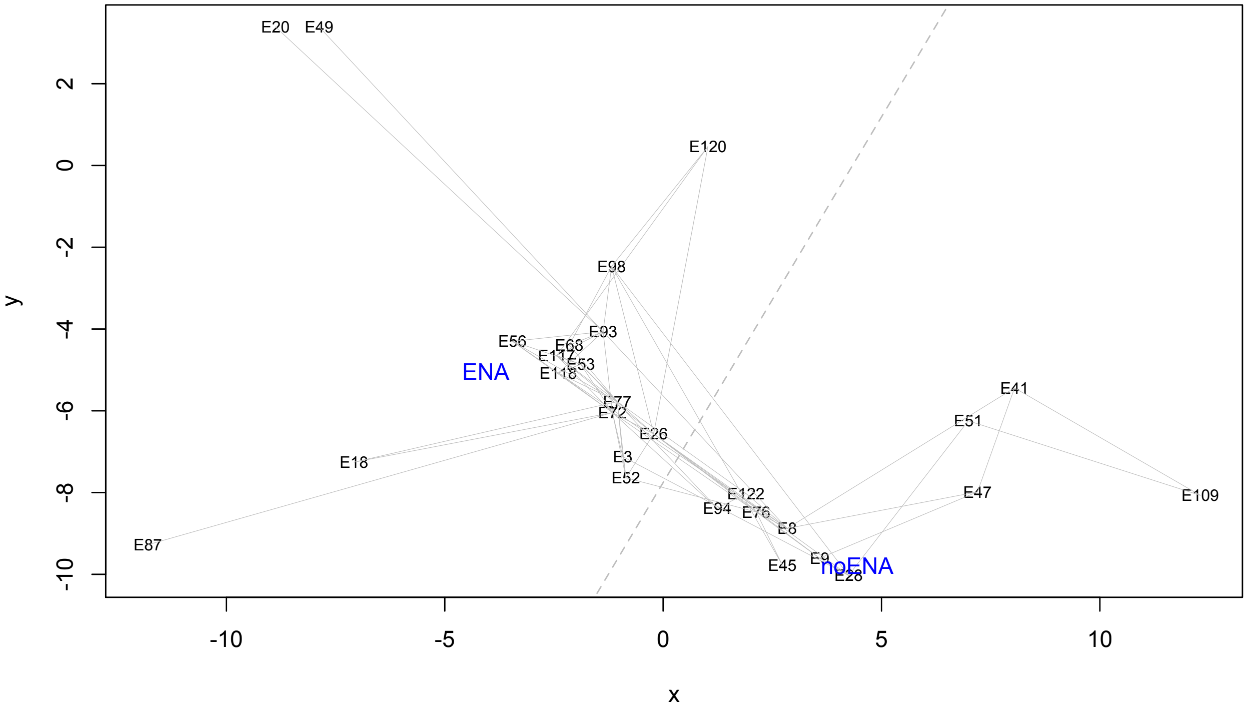

Figure 2.33: French Financial Elite - Probabilistic Plot

The higher degree nodes and the more densely tied node groups are now more conventionally positioned near the center of the plot with lower degree nodes moved to the edge (since they have 0 probability of ties with a relatively large

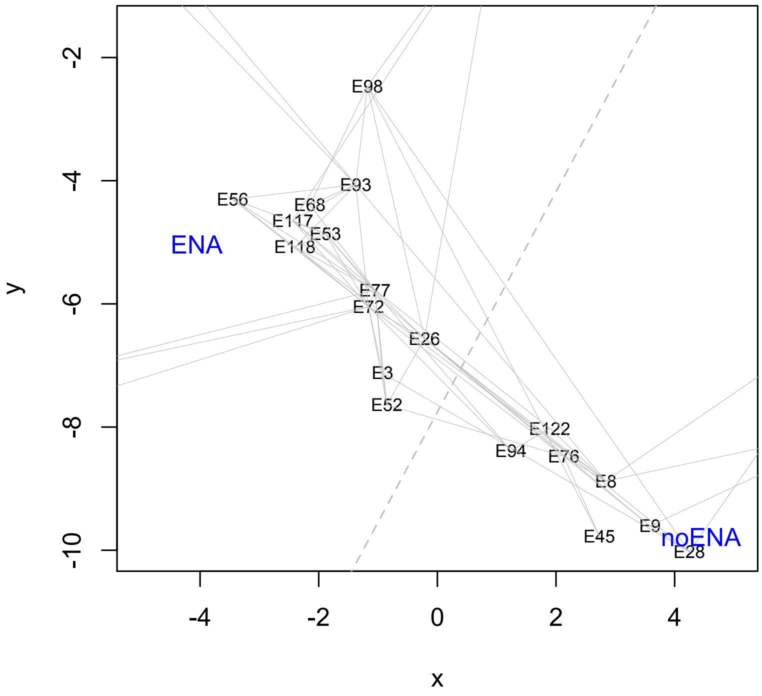

Figure 2.34: French Financial Elite - Probabilistic Plot with Magnification

number of nodes). The plot is somewhat similar to the 2-group LSM plot. Interestingly, as with LSM, it is only with the addition of the ENA covariate that we can clearly discern the 2-moiety structure. Without it, we would just as likely consider this as one tightly connected group with a few peripheral nodes.

A second example is shown in Figure 2.35. It is the Florentine families. The similarities between this plot and the force-directed plot above are quite striking. (Note that the network isolate, the Pucci family, is not shown in the LG plot since it has zero probability of a tie with any family. By convention, we would place it somewhere far from the rest of the plot.) In both, the Medicis are centrally positioned. In the LG plot, the secondary families are closer to the Medicis while the more peripheral families are further away. This makes this, perhaps, the more effective of the visualizations.

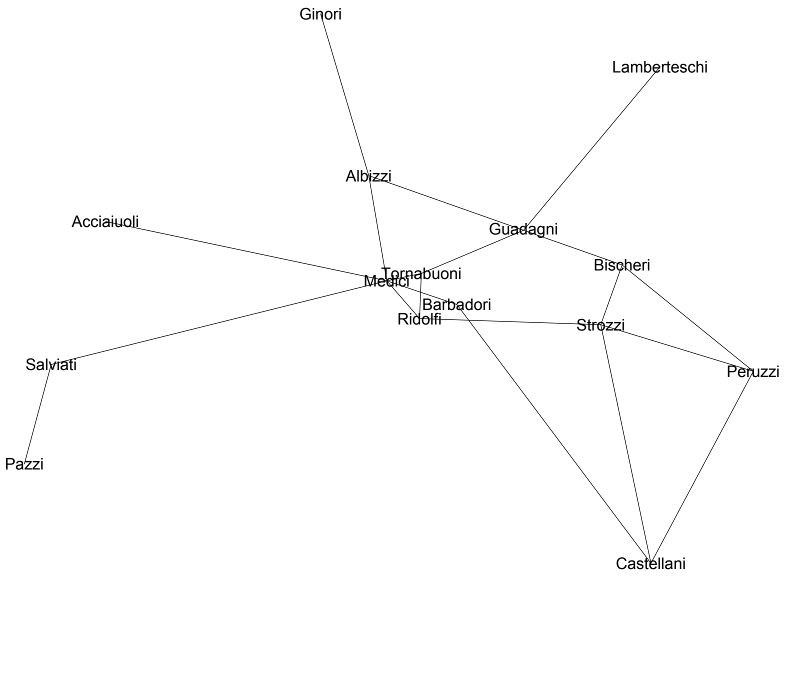

Figure 2.35: Florentine Families - Probabilistic Plot

2.5 Categorical Longitudinal Data and Markov Chain Transition Matrices

Categorical longitudinal (or transition frequency) data, in which subjects are measured on identical categorical variables at different time points, has long been analyzed using log-linear modeling. See Hagenaars (1990) [39] for a thorough introduction. Recently, distance association methods have been used in conjunction with log-linear models to relate the overall interaction to Euclidean distances. For a discussion, see De Rooij and Heiser (2005) [26]. LG can provide a method for visualizing these time-related changes.

A typical log-linear model of this sort of two-way transition table is:

$$\pi_{ij}{=}\mu\alpha_i\beta_j\text{exp(-d}_{ij}\text{(X,Y))} \tag{38}$$

where π_{ij} is the probability of a transition from category i to category j, μ is the general mean probability, α_i and βj are main effect parameters and exp(-d$_{ij}$(X,Y)) is the interaction parameter modeled as a distance association. This, of course, can be extended to three-way and other multi-way tables. For the two-way table example, if we write $\mu\alpha_i\beta_j = \frac{1}{\sum_{j=1}^{m} exp(-d_{ij}(X,Y))}$, we have a model that is essentially equivalent to the LG model. We say *essentially* equivalent because, in the log-linear model, X and Y are coordinates for variable categories at the two time points while in LG they are coordinates for subjects and categories respectively. Thus, the methods are slightly different in that the log-linear model analyzes cell counts to compute category coordinates while LG uses the raw data in the form of indicator matrices to compute both subject and category coordinates. In both cases category coordinates are computed based upon essentially the same information from the data, but to slightly different ends, with LG, in general, being the more descriptive approach.

A rich source of transition frequency data is found in political science, specifically, election and roll call data. A classic example is the 1964-1970 Swedish election data analyzed in De Rooij (2001) [25], the original source for which is Upton (1978) [65]. Table 2.7 shows this data which gives the voting results for 1651 Swedish people in 3 consecutive elections. The political parties are, from left to right on the political spectrum, the Social Democrats (SD), the Center (C), the Peoples (P), and the Conservatives (Con).

		1970			
1964	1968	SD	C	P	Con
SD	SD	812	27	16	5
	C	5	20	6	0
	P	2	3	4	0
	Con	3	3	4	2
C	SD	21	6	1	0
	C	3	216	6	2
	P	0	3	7	0
	Con	0	9	0	4
P	SD	15	2	8	0
	C	1	37	8	0
	P	1	17	157	4
	Con	0	2	12	6
Con	SD	2	0	0	1
	C	0	13	1	4
	P	0	3	17	1
	Con	0	12	11	126

Table 2.7: Transition Frequency Table for Swedish Election Data

To analyze this data with LG, it must be converted from the three-way table form to three binary indicator matrices. The matrices represent the three years with each matrix having four columns for the four political parties. The conversion is a fairly straightforward exercise. De Rooij found a 2-dimensional, 1 slide-vector model to best fit this data. The parties in this model have a diamond-shaped, or circular, configuration. It can be interpreted to suggest a dichotomy in the electorate in which the SD and C parties are together in opposition to the P and Con parties. The time element is modeled as a shift of 1 step to the political center, reflecting a significant movement of voters to the center parties from 1968 to 1970.

Since the political spectrum is typically thought of as a unidimensional construct, we chose to analyze this data first in one dimension with uLG-1 before examining the 2-dimensional models. The resulting configuration is shown in Figure 2.36. For clarity, category points are labeled off the scale and the asterisks represent the objects, which have been reduced to four groups. Overall APWL for the configuration is approximately .161 (with a low of .156 for the 1968 indicator), while classification percentages are 70.2, 72.3, and 73.0 for 1964, 1968, and 1970, respectively. For a 1-dimensional model, both of these metrics are fairly good, though perhaps just a little short of what was hoped for. This suggests that a two dimension model should probably be examined.

To continue with regard to Figure 2.36, the 1-D model, it is noteworthy that the parties are located on the scale in order from the political left to the political right. This occurred without any constraints on the model (other than dimension), so we take this to reflect that voters' perceptions of these parties were in accord with conventional political thought. Also, there is no overlapping of the party groupings; i.e., the convex hulls (intervals, in this 1-dimensional configuration) do not intersect. This shows the high level of stability in the parties over the years in question; some 79.4% of these voters did not once switch parties.[5]

The object groupings (represented by the asterisks, as noted above) are quite interesting as well. The left most point contains only the 812 persons who remained Social Democrats for all three years. The second to the left group contains all 96 persons who were Social Democrats in exactly two of the years. The second from the right group contains the 65 persons who were Social Democrats during exactly one of the years. The right most group, closest to the nine other category points, contains all 678 persons who were Center, People's or Conservative in all three years as well as all others who were never Social Democrats. This positioning is very much like a weighted Coombs unfolding, simultaneously carried out over three scales or time points.

As for the category points, the plot presents essentially a dichotomy between Social Democrat voters and non-Social Democrats. There is some shift in the latter from 1968 to 1970, but it is difficult to discern its meaning in this plot. Further investigation of this is required. Finally, a striking feature of the unconstrained plot is the relatively large space between the SD categories and those of the other parties. This is most likely due to the large number of initial SD voters: 912 of the 1651 total sample with another 60 voters becoming SD at some point. The large space allows the algorithm to position these several object points with relatively little point-wise loss, thus making it easier to limit overall APWL. This simple sort of efficiency is characteristic of the uLG-1 algorithm.

These observations lead us to De Rooij's results, making it worthwhile to test his 2-dimensional, 1 slide-vector model. To have a baseline model for comparison,

[5] This is such a dominant feature of the data, that a static, or no-change, model was tested in 1 dimension, using the 1964 party positions as the category locations for all 3 times. APWL was, again, .163 with classification percentages of 70.2, 72.6, and 71.3. Some assessment of the significance of these statistics needs to be made, but, comparing them to the unconstrained results, the lower classification success for 1970 may justify rejecting the static model. This is particularly so in light of the results for a constrained 1-dimensional model with 1 shift (in 1970) of the Social Democrats and Conservatives each 1 step to the center. APWL was .161 with classification percentages: 70.7, 73.6, and 73.5.

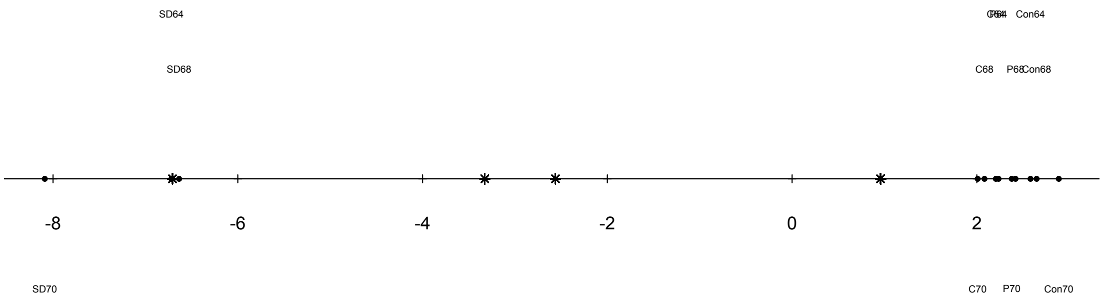

Figure 2.36: Swedish Election Category Points in 1D

we first ran an unconstrained uLG-1 model in 2 dimensions, shown in Figure 2.37. APWL is .036 with classification percentages of 92.2, 96.3 and 97.9. These improvements over the 1-dimensional model are to be expected given the greater ease of fitting LG models in higher dimensions and may signify nothing beyond that. That is to say, we appear to be overfitting the data. Despite the increase modeling accuracy, the configuration seems to show little important structure. The convex hulls of the party clusters do not intersect, reflecting the stability of the data noted above, but this was seen in one dimension, also. The Center party being most centrally positioned and in the tightest cluster may reflect that it is the only party to have gained votes from 1964 to 1970, but this is something of a conjecture.

An interesting feature of the plot is that, though there are 1651 objects in the dataset, there are only 49 object points. That is because there are only 49 actual party transition groups. (There are 64 of these possible, of course, but in the actual data 15 of these have zero counts.) The objects in each of these groups have essentially identical coordinates. Thus, each object point can be thought of as a group point weighted by the number of objects in the group. This observation may be helpful in later work in constructing a more efficient algorithm.) We have labeled some peripheral points in the interest of studying this aspect of the LG method. Note that they are all party-switching profiles with very few members. (The membership numbers are given in parentheses. Thus, they can be moved away from the main configuration without large increase in loss. Finally, note the asterisk points which are the no-party-switching object classes. They are located in a diamond shape very similar to that found by De Rooij, though not with the same ordering of parties. Despite the ordering, this configuration reinforces that it is worthwhile to test the 1 slide-vector model with constrained LG.

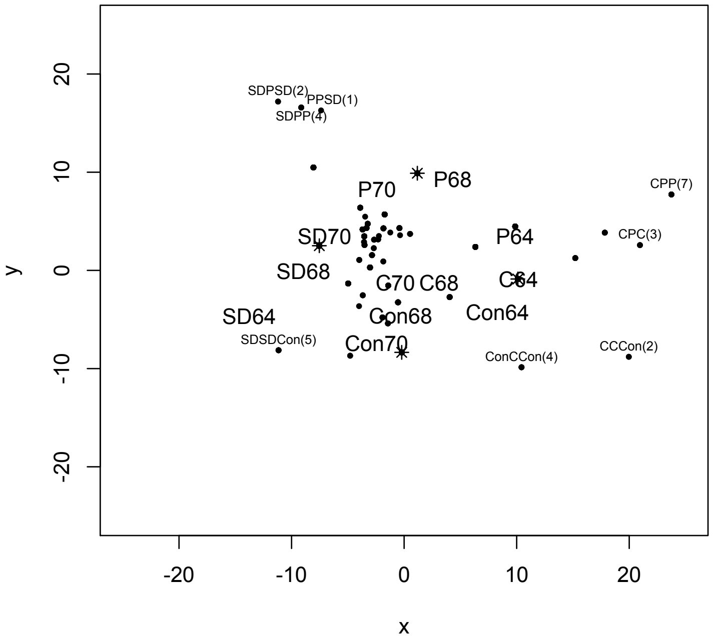

Figure 2.37: Swedish Election Object & Category Points

With categories fixed in the two-dimensional, 1-slide slide-vector configuration (in which the 1964 categories also represent the 1968 categories), constrained LG was run until average coordinate change was .000.[6] Figure 2.38 shows the plot which was produced. APWL is .222, which seems rather high. Note that it is, in fact, higher than the APWL for the unconstrained 1D plot. Interestingly, however, classification accuracies[7] are much higher: .900, .953, and .941. At first, the high APWL seems to be an indication of the degree of the constraint we have imposed on the model. But considering the high classification rate and our work on the IMT, we realize that it is solely a function of the chosen scale for the category constraints. If we dilate our configuration by a factor of 10, thereby producing a geometrically equivalent configuration with identical classification accuracy, we obtain an APWL of .041, which is quite comparable to the unconstrained 2-dimensional plot. Experimenting with dilation factors shows this APWL to be essentially optimum for this configuration. This is a subject we will return to in later sections.

An obvious feature of this plot is that the object points seemed divided into two groups along an axis parallel to the slide vector - i.e., along a line that divides the left and right halves of the political spectrum - with some points from each group arrayed around the periphery of the category configuration. Examining some of these points give us some insight into the structure of the plot. Points close to the axis line are groups with voters switching across the political spectrum. There are relatively few of these types of voters and generally larger numbers of voters in the peripheral groups, which involve switches within the same half of the spectrum. This highlighting of the importance of the political left-right axis is probably the most important contribution of this model over the 1D model, which clearly divides the parties much differently. One can see from this process that

[6] It should be noted, if it has not yet been made clear, that the constrained LG algorithm uses a fixed-point convergence criterion, unlike uLG-1 which uses an APWL target. With constrained categories, there is little concern that the algorithm will provide reflected, translated, or rotated configurations from iteration to iteration.

[7] Recall that these are based on maximum model category probability.

many exploratory hypotheses about this data can be studied using constrained LG.

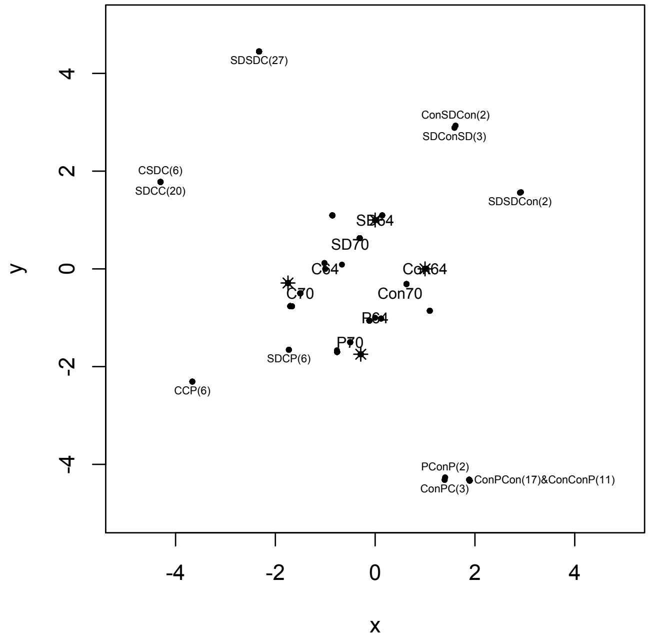

Figure 2.38: Swedish Election Data - Slide Vector Model

A natural extension of the use of LG with longitudinal data is to use the method to visualize discrete-time Markov chain transition matrices. For a Markov chain with one-step transition matrix, P, we compute n-step state probabilities by taking successive powers of P until we reach the chain's steady-state (assuming convergence of the sequence of powers of P). Viewing the states as objects and each time step as a variable, each transition matrix can be considered a fuzzy indicator matrix and this collection of indicators can be input into uLG-1. The resulting configuration can give an interesting view of the progression of the Markov chain.

Consider the matrix shown in Table 2.8 which is used as an instructional example in Nelson (1995) [50]. To give some context, it can be thought of as a transition matrix to model basic activity on a particular website, state 1 being

S1	0	.95	.01	.04
S2	0	.27	.63	.10
S3	0	.36	.40	.24
S4	0	0	0	1

Table 2.8: Markov Chain One-Step Transition Matrix

login, state 2 being a search, state 3 a scrolling or reading, and state 4 a log off. After 43 steps, the chain converges (to 4 decimal places) to the matrix in Table 2.9; i.e., all users are log offed. This data, four states as objects measured on 43 variables, each with a 4 × 4 fuzzy indicator matrix, was input into uLG-1 for a 2-dimensional model. The resulting configuration is shown in Figure 2.39. APWL is .0018 (after 1500 iterations), indicating that 2 dimensions is adequate for fitting the probabilities. The (trivially) transient log on state moves quickly at first, then slowly to infinity, no return to it from any other state being possible.[8]States 2 and 3 show fairly quick, simultaneous convergence to the steady-state. State 4, the log off, an absorbing state, moves steadily toward the active states.

S1	0	0	0	1
S2	0	0	0	1
S3	0	0	0	1
S4	0	0	0	1

Table 2.9: Steady-State Matrix

Next, we modify the matrix slightly by adding a fifth state for log off by timing-out. (Like state 4, it is an absorbing state.) The one-step transition matrix is shown in Table 2.10. The chain converges (again, to 4 decimal places) in 49 steps

[8] Note that we could have removed state 1 *as a category* - that is, removed the first column from each indicator since they are all columns of zeros. Generally, this is advisable to increase the efficiency of the algorithm and because such categories are of little analytical interest. The category was left in here for demonstration purposes.

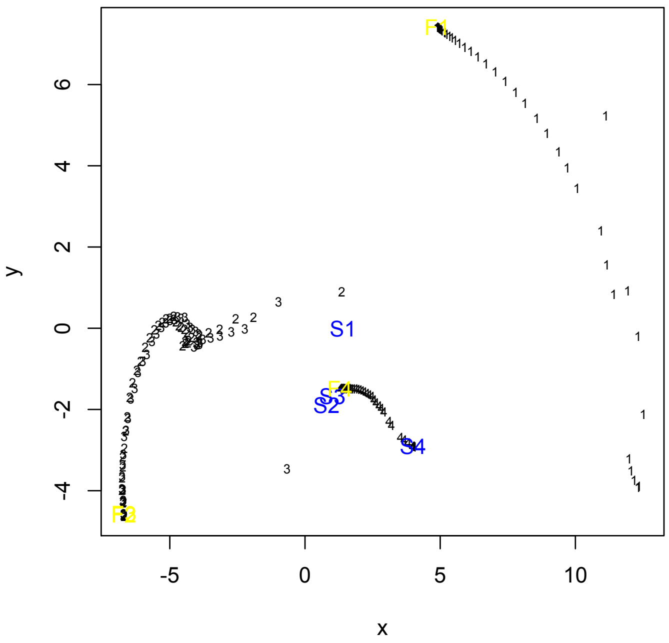

Figure 2.39: Markov Chain N-Step Probabilities

to the steady-state matrix shown in Table 2.11. For this run of two-dimensional uLG-1, we will eliminate the state 1 *category* (the column of zeros) from each of the 49 n-step probability matrices as discussed in the footnote to the above paragraph.

S1	0	.95	.01	.03	.01
S2	0	.27	.63	.09	.01
S3	0	.36	.40	.23	.01
S4	0	0	0	1	0
S5	0	0	0	0	1

Table 2.10: 5-State Markov Chain One-Step Transition Matrix

S1	0	0	0	.9341	.0659
S2	0	0	0	.9417	.0583
S3	0	0	0	.9483	.0517
S4	0	0	0	1	0
S5	0	0	0	0	1

Table 2.11: 5-State Steady-State Matrix

The resulting configuration is shown in Figure 2.40. APWL is .0059 after 1495 iterations. Again, states 2 & 3 are positioned together near the center of the plot. The absorbing states, 4 & 5, converge directly to them, though state 4 comes much closer, the log off being far more probable than the time-out. The convergence pattern of states 2 & 3 are each marked by a discontinuity which suggests an area for further investigation of the chain. Because of this and the slightly high APWL of the 2D plot, a uLG-1 model was run in three dimensions. APWL is .0015 (after only 581 iterations) indicating somewhat better fit. The 3D plot (Figure 2.41) shows some similar patterns to the 2D. States 2 & 3 (labeled by numbers) are close to each other near the center of the plot. The absorbing states 4 and 5 converge toward them as before. (As categories, they are marked by o's and t's, respectively, for log *O*ut and *T*ime-out, due to the labeling procedures of the `scatterplot3d` function.) Notice how the time-out becomes less likely at first, then more likely as it converges. Of course, what is most noticeable is that the discontinuities in the convergence patterns of the state 2 and state 3 categories (labeled *S*earch and *R*ead, respectively) are here even more pronounced, to the point where they would appear to more significantly affect model probabilities. A computational analysis of this would likely be of some interest to those studying this chain.

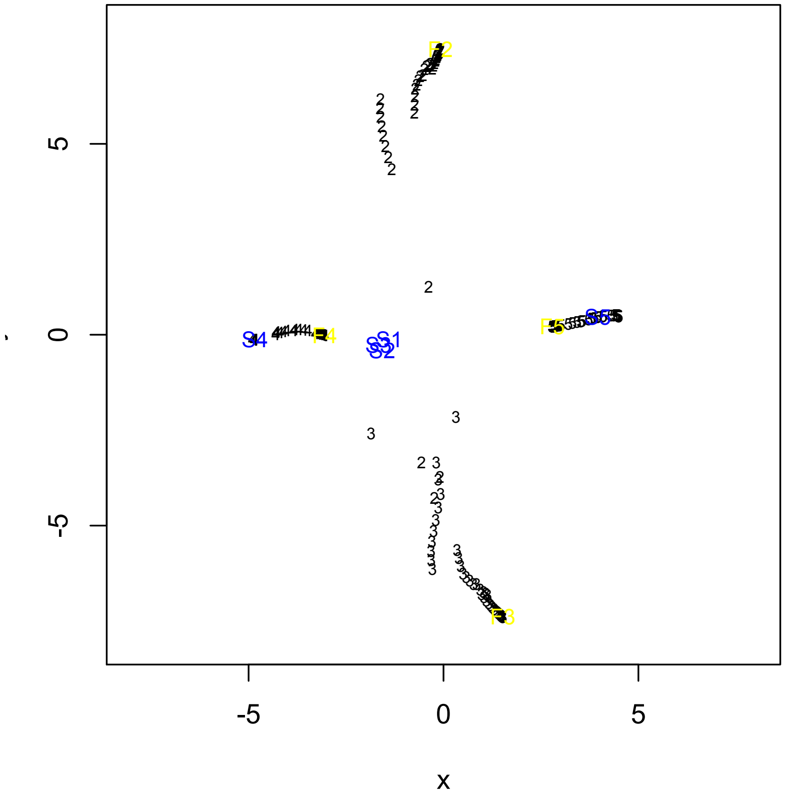

Figure 2.40: 5-State Markov Chain N-Step Probabilities

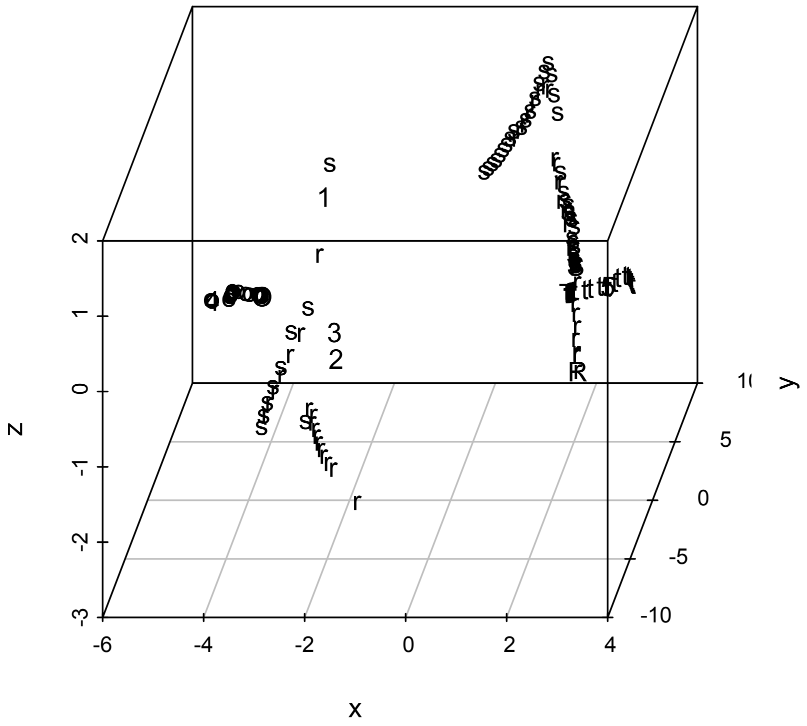

Figure 2.41: 5-State Markov Chain N-Step Probabilities - 3D

CHAPTER 3

Convergence

From our work to this point, we see that to better understand the creation of uLG-1 plots, an analysis of the convergence properties of the LG algorithm is needed. In carrying this out, we will restrict our attention, at the outset, to uLG-1 with binary indicator matrices. The analysis involves two key questions: The first is whether, for a given dataset, there is a stationary point of the algorithm; i.e., some configuration that is a global minimum for the APWL function. The second somewhat related question is what is the rate of convergence.

The IMT gives us some insight into the first question. We know that if the data can be plotted so that all objects are in the Voronoi cells of the categories they belong to (i.e., we have what we refer to as 100% classification accuracy), then larger and larger dilations of such a configuration will produce lower and lower APWL's. Though from a practical point of view, we obtain quite satisfactory plots in this way, the APWL function in this case has no global minimum and the algorithm has no stationary point.

This situation is similar to what is known, in connection with logistic regression, as complete separation of the data. In a classic paper, Albert & Anderson (1984) [3] show that, if there is complete separation of the data, the MLE of the logistic regression parameters does not exist, in the sense that, as in LG, they are attained at infinity, on the boundary of the parameter space. We will adopt this terminology and refer to data that satisfy the condition of the IMT as completely separated.

Given this result, it is of interest to determine if a particular dataset can be completely separated in low (i.e., 2 or 3) dimensional space. If so, the LG algorithm need be run only until 100% classification accuracy is attained or, to be more precise, only until the categories are positioned so that the number of separating Voronoi cells[1] equals the number of distinct object profiles, although this latter condition is more difficult to test during the operation of the algorithm. Then, dilation will give APWL to any desired cutoff.

To determine whether a dataset is completely separable is, in general, a rather complex problem. We will restrict our attention to 2-space and first consider some particular cases, then comment on the general case. To begin, assume all of the variables are dichotomous. It is obvious that, if $j =$ the number of variables, then for $j = 1$ or 2, the data is completely separable. However, if $j \geq 3$ and all possible object profiles are found in the data, they are not. This is shown as follows. Note that each dichotomous variable places 2 category points in the configuration. The Voronoi cells of these are the half planes determined by the line bisecting the segment connecting the category points for the variable. The separating cells are the intersection of these half-planes. Figure 3.1 gives an illustration for 3 dichotomous variables.[2] Notice there are 3 bisecting lines resulting in 7 separating cells, 1 bounded and 6 unbounded. This illustrates a result of Coombs & Kao (1955) that, in 2-space, n separating lines will create:

$$(39) \qquad \tau(n\ ,2) = \textstyle\sum_k^2 \binom{n}{k}$$

profile cells (of which $2n$ are unbounded). For n = 3, τ(3,2) = 7 and, in general, for n $\geq$ 3, τ(n,2) $<2^n$, which is the number of possible distinct object profiles.

[1] By separating Voronoi cells, separating cells or profile cells, we mean an intersection of category Voronoi cells, each from a different variable, corresponding to an object profile classifying the object in those categories.

[2] We have used the `R` package `deldir` to draw the Voronoi cells. `deldir` stands for Delaunay triangulation and Dirichlet tessellation, the latter being the process of drawing the Voronoi cells.

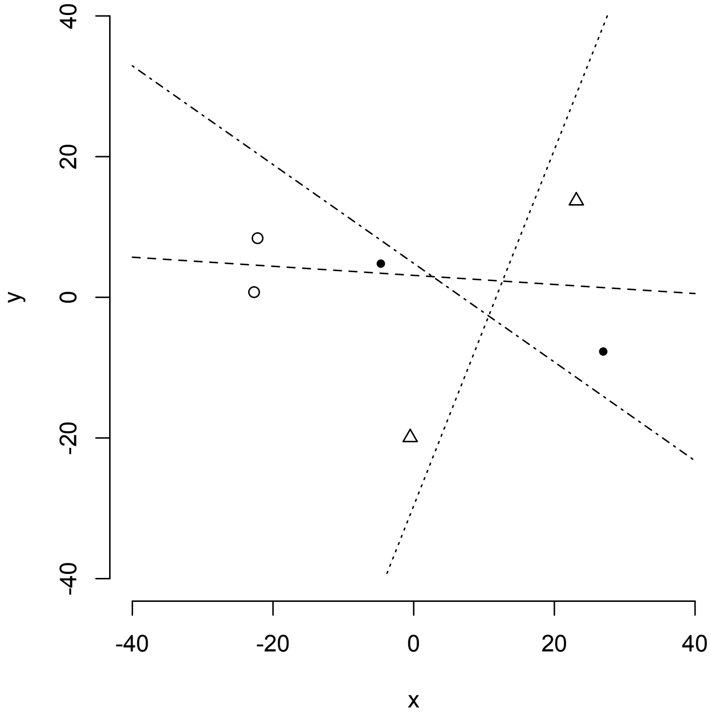

Figure 3.1: Separating Cells for 3 Dichotomous Variables

To study the workings of the LG algorithm, we created a dataset with 100 objects randomly classified on 3 dichotomous variables with all 8 possible object profiles appearing. The uLG-1 plot is shown in Figure 3.2. The objects are grouped into 8 distinct object points. Notice that there are 2 object profile points in the 2-1-1 (2-a-A) profile cell. The point closest to the origin, which contains 6 misclassified objects, is the 2-2-1 (2-b-A) point for which there is no profile cell. APWL for this configuration is .041 after 15000 iterations of uLG-1.[3]

[3] Interestingly, dilation by a factor of 10 lowers the APWL to .0205 and this appears to be very near the global minimum for this configuration. We see here that, for classification near 100%, dilation can be effective for lowering APWL, though not to 0. For less accurate classification, dilation tends to increase APWL.

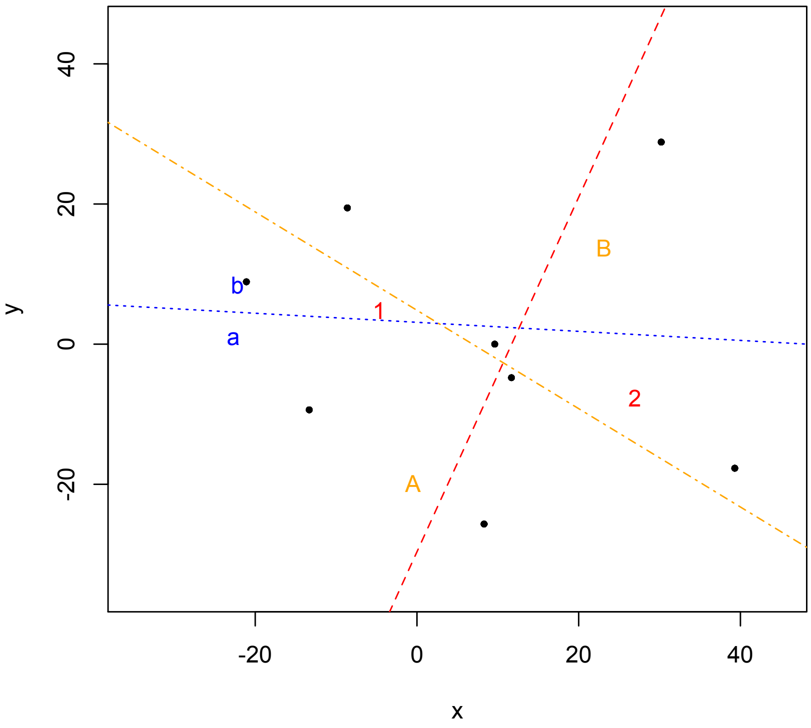

Figure 3.2: Three Dichotomous Variables with 8 Profiles

Removing the 6 misclassified objects, giving us completely separated data, results in the plot in Figure 3.3. Again, there are 7 separating cells and, now, all objects are correctly classified. APWL is .0009 after 3335 iterations and .0002 after 15000. Dilation of the .0009 configuration by a factor of 10 produces a an APWL of 1 × 10^{-26}. It is important to note that the pictured and the dilated configurations have congruence coefficient and distance correlation of 1.

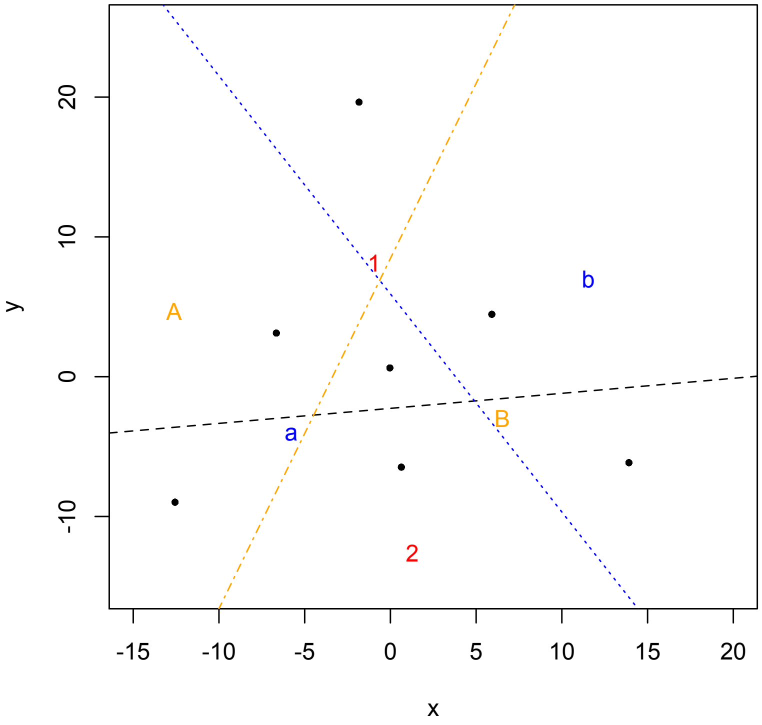

Figure 3.3: Three Dichotomous Variables with 7 Profiles

Applying our previous observations about algorithm operation, we note that, at 19 iterations, a configuration with .063 APWL gives 100% classification. Dilation of this by a factor of 10 gives an APWL of 1 × 10^{-8}, far lower than even a 20000 iteration run of the algorithm and in a miniscule fraction of the time.

For polytomous variables, determining the number of profile cells that can be created in 2-space seems, at first, to be a straightforward problem. Figure 3.5 shows that, for 2 variables each with 3 categories, a configuration giving 9 profile cells, and, hence, complete separation, can be created. In fact, we can see from Figure 3.5 that, in general, for 2 variables, if variable 1 has m categories and variable 2 n categories, then the category points can be linearly positioned along the coordinate axes to create a checkerboard pattern with nm separating cells, hence, complete separation, as is shown in Figure 3.4. This can easily be seen to generalize to up to n variables in n-space. Of course, whether LG will create such profile cells is a different question. In Figure 3.6, we see a case where it does for a synthetic dataset of 100 objects with profiles generated at random on two 3-category variables in which all 9 object profiles appear. The plot is created after 18 iterations of the algorithm. A factor-of-10 dilation gives APWL of 1×10^{-7}.

For more than two polytomous variables in 2-space, it follows from the Coombs-Kao formula (35) that the number of possible object profiles will be greater than the number of possible profile cells. We conjecture that this is true in general; i.e., for more than n polytomous variables in n-space. In that case, there can be no complete separation for such data. In the terminology of Albert & Anderson, data of this sort may be said to be overlapping. For logistic regression, Albert & Anderson show that, for such data, parameter estimates can be found; i.e., the likelihood function has a global maximum. Whether this is true of LG is an open question, but our work with dilations of less-than-100%-classification configurations suggests that it is.

A third result of Albert & Anderson is that, for data that has quasicomplete separation, again, parameter estimates do not exist. In LG, quasicomplete separation means, roughly put, that some object points will be on the borders of category Voronoi cells. A simple example of this is a 4-object dataset with the

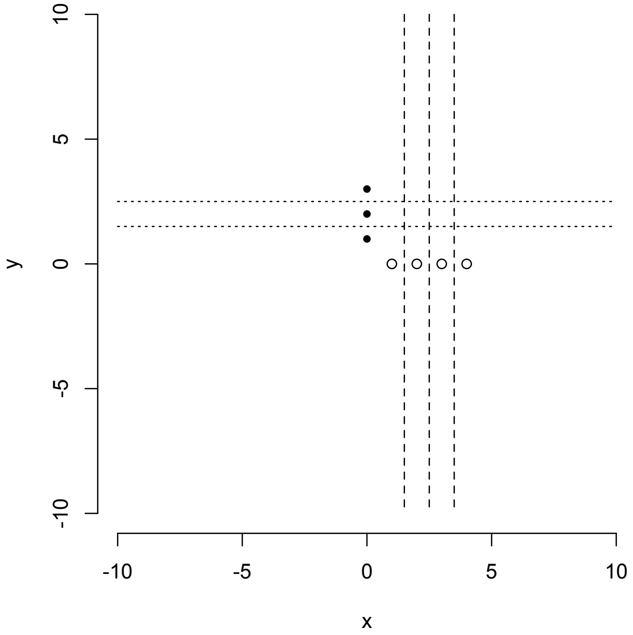

Figure 3.4: Separating Cells - Three-by-Four Checkerboard Pattern

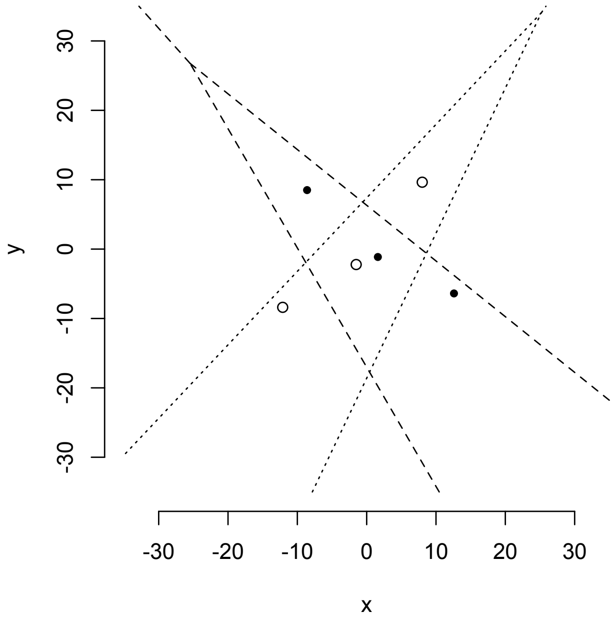

Figure 3.5: Nine Separating Cell Configuration for Two 3-Category Variables

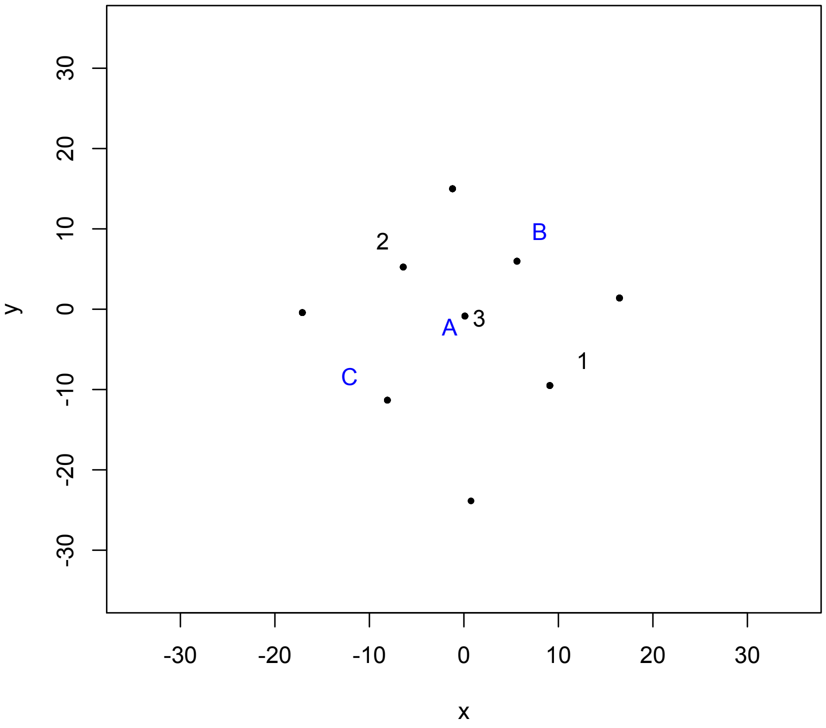

Figure 3.6: Two 3-Category Variables with 9 Profiles

following indicator matrix to be plotted in 2-space:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

Obviously, as we have commented on previously, we can draw a configuration with APWL arbitrarily close to 0 by placing object 4 at the origin and the other 3 objects exactly at their category points all on a large circle centered at the origin. Thus, as with complete separation, no configuration giving a global minimum APWL can be found.

The uLG-1 algorithm takes essentially this approach. Figure 3.7 shows successive configurations found through 20000 iterations of the algorithm. The open dots show the final configuration which has APWL of 1.8×10^{-5}. Figure 3.8 shows the sequence of APWL's. After dropping quickly, they move only very slowly toward the limit of 0.[4] Figure 3.9 is a magnification showing this process more clearly.

Of course, by placing the 3 binary object-category points on a circle of radius, say, 1000, we can, without the aid of the algorithm, instantly create a much lower APWL configuration. This brings us directly to the question of convergence rate. The performance of the algorithm on the 4-object quasiseparated data above illustrates a result of De Leeuw (1988) [12] regarding the `smacofRect` MDU algorithm at the heart of uLG-1. In this seminal paper, De Leeuw showed that the algorithm

[4] Interestingly, weighting the data by adding a large number of objects with one of the binary profiles does not seem to change the operation of the algorithm. That is, that object is not moved further away more quickly to decrease APWL.

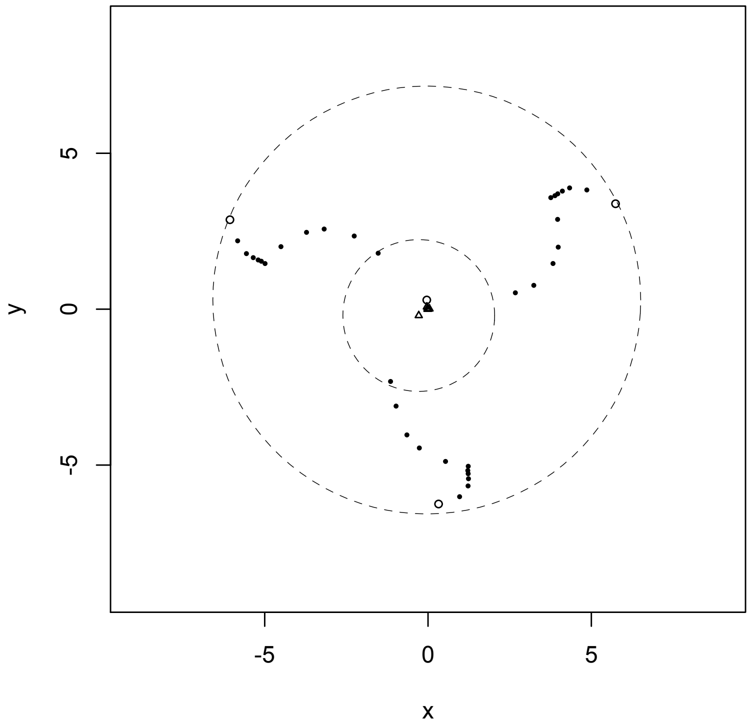

Figure 3.7: Successive Circular Configurations

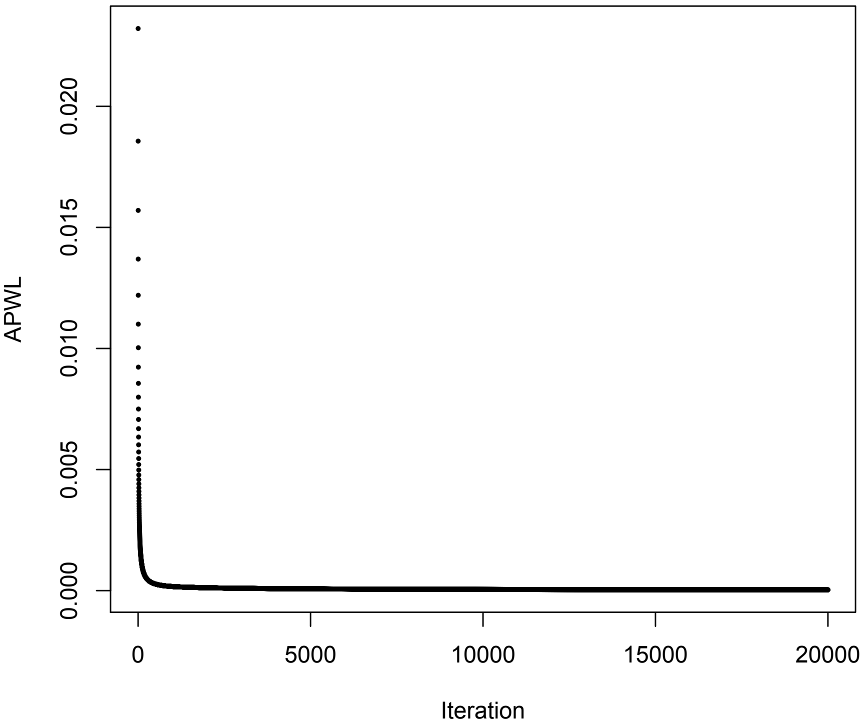

Figure 3.8: Sequence of APWLs for Circular Configurations

converges linearly to a minimum stress configuration with rate given by the largest eigenvalue not equal to 1 of the second derivatives of the `smacof` update evaluated at the stationary point. In terms of LG, in which APWL arises primarily from MDU stress, if the sequence of APWLs generated by the algorithm, x_k, converges to some value, L, we have:

$$\lim_{x\to\infty} \frac{x_{k+1}-L}{x_k-L} = \lambda \leq 1. \tag{40}$$

Further, in almost all cases λ is very close to 1, which is referred to as sublinear convergence. De Leeuw conjectures (personal communication, 2013) that, if no stationary point exists; i.e., if APWL is minimized by moving objects or categories to ∞, then convergence is, in fact, sublinear. Figure 3.10 shows the ratio of APWLs for the iterations of the 4-object quasiseparated data which, along with a number of other examples we have examined, seems to strongly support the conjecture.

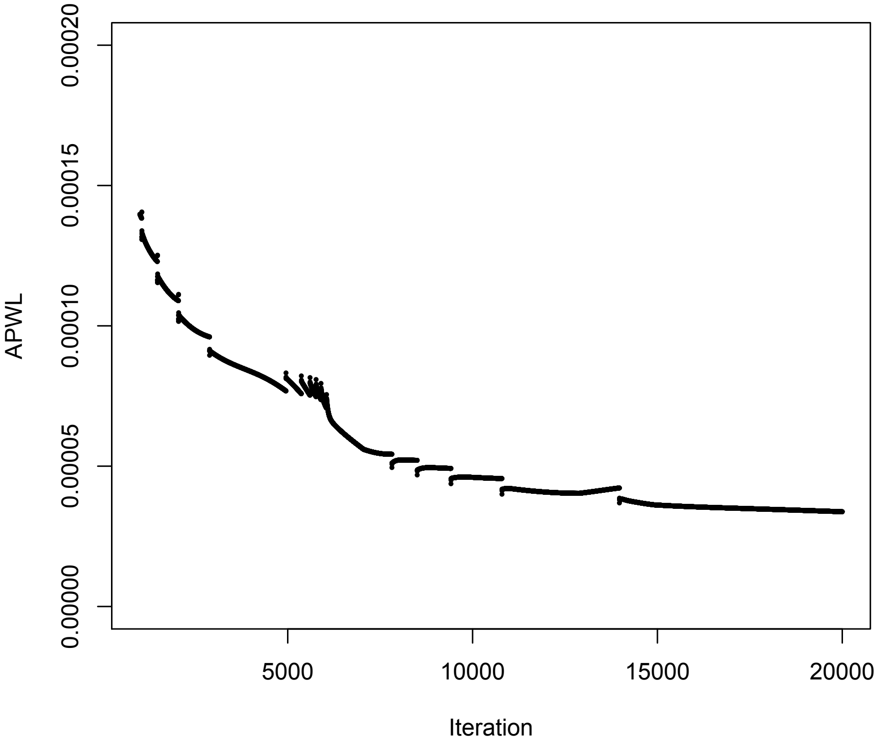

Figure 3.9: Sequence of APWLs for Later Iterations

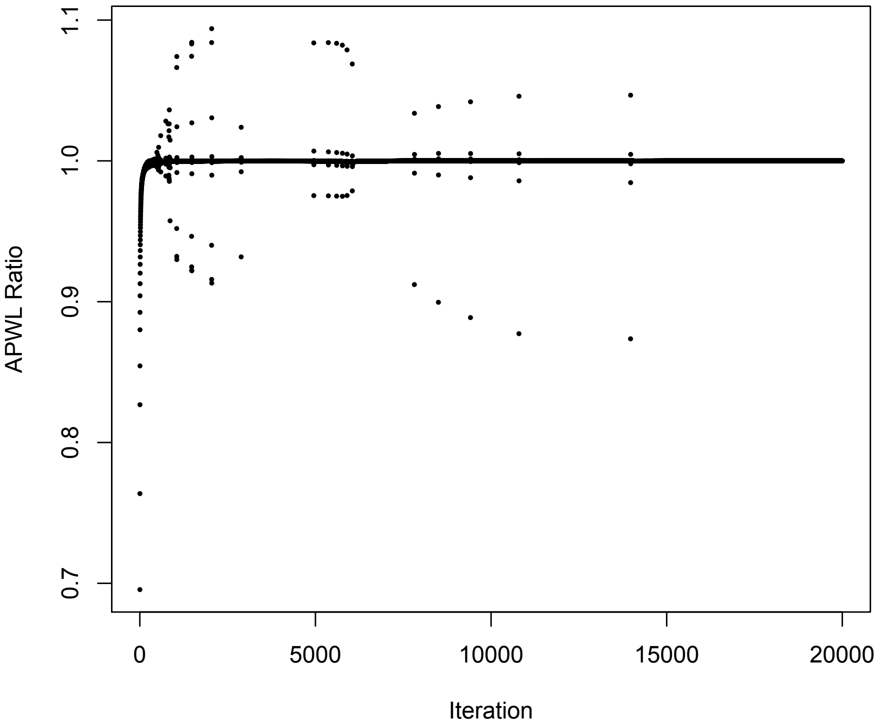

Figure 3.10: APWL Ratios

CHAPTER 4

Algorithm Acceleration

As we have seen, with even moderately sized data sets, if we select a desired APWL target (which may by chance be lower than what can be obtained) and have the uLG-1 algorithm run until it is met or until some large iteration limit is reached, considerable computing time may be required. For example, the 1495 iterations of uLG-1 used to produce the two-dimensional 5-state Markov chain model ran for over 2 hours and datasets with many variables, such as typical roll call data, may run much longer. This is, of course, a major impediment to using LG. Runs of the LG algorithm which include bias parameter computation will likely be even more time-consuming.

From the IMT, we see that a possible time-saving approach would be to run just enough iterations of the algorithm to produce a reasonable approximation to the stationary configuration, then to dilate it until optimum APWL is reached. Our discussion of the 2-dimensional slide-vector model of the Swedish Election data gives an example of how this method might work. However, we cannot determine in advance what a sufficient number of iterations might be and may end up undershooting the number, which will result in a sub-optimum configuration, or overshooting, which will result in wasted computing time. We must also find an optimum dilation factor, an essentially heuristic process which, though not particularly difficult, still adds to our computing time.

Fortunately, there is an approach which essentially carries out this process, but overcomes the difficulties mentioned above. It is known as Minimal Polynomial

Extrapolation (MPE). We have had some success applying MPE to accelerate the algorithm. MPE allows us to compute configurations equivalent to those of the maximum-iteration approach with far fewer iterations of the algorithm and consequent saving of system time. It does this by essentially projecting forward from a few iterations to calculate an approximate stationary point of the algorithm. We do not need to guess at a target APWL or test if some sub-optimal configuration is close enough in congruence to be dilated. And, the sub-linear convergence property of uLG-1 discussed above allows MPE to function well even if, in theory, some object or category points should be moved to ∞. We provide a brief outline of MPE here. An excellent and thorough discussion of the method is found in Smith et al. (1987) [59].

We suppose that we have a sequence of N-dimensional vectors of the form

$$(41) \qquad \mathrm{x}_{j+1} = \mathrm{Ax}_j + \mathrm{b},\ \mathrm{j}=0,1,2,\ldots$$

where A and b are fixed, but unknown. Assume, for the moment, that all eigenvalues of A are less than 1 so that the iterative sequence will converge to a unique fixed point. MPE allows us to find that fixed point without computing A and without inverting an $N \times N$ matrix. The procedure, for the details of which the reader is referred, again, to the very elegant presentation of Smith et al. cited above, is as follows: We first compute the difference vector

$$(42) \qquad \mathrm{u}_0 = \mathrm{x}_1 - \mathrm{x}_0,$$

then consider the monic polynomial of least degree, or *minimal polynomial*, P(z), such that

(43) $\mathrm{P(A)u_0} = 0.$

Suppose the degree of P is k. We then form *k+1* difference vectors

(44) $\mathrm{u}_j = \mathrm{x}_{j+1}$ - x_j, j=0,1,2,...k

and column-bind them into a matrix, U. It can then be shown that the non-leading coefficients of P, which we denote by the vector c (recall that P is monic so the lead coefficient is 1) are given by

(45) $c = \text{-}\mathrm{U}^{+}\mathrm{u}_k$

where U^{+} is the Moore-Penrose generalized inverse of U. Further, it can be shown that the fixed point or limit of the sequence, denoted s, can be computed as

(46) $s = \frac{\sum_{j=0}^{k} c_j x_{m+j}}{\sum_{j=0}^{k} c_j}$

for any k+1 consecutive terms of the vector sequence, x_m,. . .,x_{m+k}.

To this procedure, three clear objections, two general and one specific to LG, are sure to be raised. The first general objection is how can k be determined. The answer is that, in general, it can't be. In practice, we proceed by approximating k by using vector subsequences of different lengths to construct the matrix, U, and to ultimately compute s, the limit vector. Smith et al. report that this approach has been found to lead to good approximations of s. The second general objection is how to proceed if the vector sequence does not converge for all elements of the vector. Non-convergence for uLG-1 might occur, for example, if the configurations returned at each iteration of the algorithm are not uniformly centered and oriented. Fortunately, the SMACOF algorithm, which computes the coordinates of the configurations, always returns centered category configurations.[1] This has proved to be a sufficient guarantor against non-convergence induced by translations or rotations of the configurations.[2] In general, non-convergence might also occur if points of the vector are moving to ∞. In such cases, Smith et al. give methods to transform the sequence to one that converges to an *anti-limit* and give ways to interpret these. We have not yet encountered this situation with our LG-related experiments with MPE. This is likely because, although for some datasets uLG-1 will ideally move points toward ∞, convergence of the algorithm in such cases is so slow that MPE will compute finite coordinates.

The LG-specific objection pertains to how MPE can be applied to LG since it does not appear to generate vector sequences by linear iterations. First, it is true that the main output of uLG-1 are two configuration matrices, one for objects the other for categories. So, to apply MPE to the algorithm, we simply vectorize these matrices, obtain the limit vector, which is, after all, given by the coordinate-by-coordinate limits, then reconstruct the configuration matrices. Second, it is also true that the functional iterations of uLG-1 are not linear. However, as discussed by Smith et al., and in greater detail by De Leeuw (2008) [20], if the sequence-generating function, F, can be approximated by a linear function in a neighborhood of the fixed point, s, MPE can still be applied, since we have, for all x in such a neighborhood

$$(47) \qquad \mathrm{F}(\mathrm{x}) \approx s + d\mathrm{F}(s)(\mathrm{x}\text{-}s) = d\mathrm{F}(s)\mathrm{x} + (\mathrm{I}\text{-}d\mathrm{F}(s))s \ .$$

[1] This is because, in SMACOF as it is applied in uLG-1, each iteration gives a configuration of the form $\mathrm{X}_{k+1} = \mathrm{V}^{+}\mathrm{B}(\mathrm{X}_k)\mathrm{X}_k$ where the matrices V^{+} and $\mathrm{B}(\mathrm{X}_k)$ are symmetric matrices with row sums equal to 0. In particular, for $\mathrm{B}(\mathrm{X}_k)$ the diagonal elements are equal to the negative of the sum of all other row elements. The matrix V^{+} has a slightly different structure; for our applications V^{+} is a block-diagonal matrix with blocks of the form $\mathrm{n}_1{}^{-1}(\mathrm{I} - \mathrm{n}^{-1}11^{\mathrm{t}})$. See De Leeuw & Mair (2009) [24] for details of the derivation.

[2] Though it is not commented on explicitly, this appears to have been the experience, as well, of the researchers who conducted the leading published study of SMACOF with MPE, Rosman, et al. (2008) [56].

Of course, in such a case an exact solution for s cannot be guaranteed, but excellent results can still be obtained. Inspection of the functions that comprise the uLG-1 algorithm, show them to be sufficiently well-behaved for (47) to apply. This is fairly clear for the distance and probability calculations that are made at each iteration. With regard to the unfolding process, we see from a review of de Leeuw & Mair (2009) [24], that the `smacofRect` algorithm computes unfolding configurations by repeated minimizations of a simple quadratic function. This can be assumed to be well-behaved also, so we can apply MPE to the LG algorithm.

The MPE procedure, then, is to run a small number of iterations of uLG-1, saving the resulting configurations for each, then vectorize the configurations as discussed above, compute the one-step difference vectors u_j (42) and the vector c by (45), then compute s using (46). We have used the simple R code for this found in Loisel & Takane (2011) [46].[3] The vector s is then formed into the configuration matrix which is then evaluated for APWL. All values of k from 2 up to the length of the vector sequence are used to compute a vector, s. Subsequences from the beginning and the end of the generated sequence are used, with the lowest APWL configuration taken as the output. The R function `mpeLG1` found in the supplementary file carries this out.

The results, so far, have been very promising. For the 5-state Markov chain (which, as noted above, took nearly 1500 iterations and over 4 hours to run on `UnbiasLG`), 30 iterations of `UnbiasLG` were used for `mpeLG1`. After a total running time just over 4 minutes, a configuration was found with APWL = .0101 (compared to .0059 using the unaccelerated algorithm). This configuration was computed using 14 of the 30 vectors in the generated vector sequence (or, put another way, by approximating k, the degree of the minimal polynomial, to be 13). It is plotted in Figure 4.1 and can be seen to be strikingly similar to the

[3] It uses the `ginv` function from the `MASS` package in R.

original uLG-1 plot in Figure 2.40.[4] That the second one was computed with a nearly 6000% decrease in computing time is somewhat astounding.

Next, we computed a configuration for the French Financial Elite network using `mpeLG1`. Recall that, for the configuration in Figure 2.31, APWL is .1343. This was obtained at the max-iteration termination (2500) after over 2 hours of system time. Using `mpeLG1`, a configuration with APWL of .0891 was computed in just under 10 minutes. This involved a vector sequence of length 200, the last 145 of which were used to compute the limit vector. The configuration is shown in Figure 4.2. Its structural similarity to the original plot is obvious.[5]

To try MPE with fuzzy indicator data, we used the method to compute a configuration for the LA Neighborhoods data used in the section on multinomial regression. The uLG-1 configuration, with APWL of .0223, was produced in only 65 seconds of system time. With MPE, a .0346 APWL configuration (Figure 4.3) was produced in 2.6 seconds using the last 68 vectors of a 100-vector sequence. But for a rotation, it is essentially identical to Figure 2.19, the uLG-1 plot (congruence coefficient of .988, distance correlation .954). It is pictured below.

From considering the above process, a technique for improving the estimate

[4] Indeed, the congruence coefficient between the 2 configurations is .993, with distance correlation of .972. In MDS or MDU, the generally accepted approach for comparing configurations is to compute both of these statistics. (Simple examples to show that it can be misleading to rely on the correlation alone are given in Borg & Groenen (2005) [6]. The congruence coefficient, c(X,Y) , between the configurations given by coordinate matrices X and Y is given by

$$\text{c(X,Y)} = \frac{\sum_{i<j}^{j} d_{ij}(X) d_{ij}(Y)}{((\sum_{i<j}^{j} d_{ij}^2(X))^{.5}(\sum_{i<j}^{j} d_{ij}^2(Y))^{.5})}.$$

Notice that by the Cauchy-Schwartz inequality, $0 \leq$ c(X,Y) ≤ 1 and c(X,Y) $= 1$ if X and Y are perfectly geometrically similar. This similarity can, in general, include difference by dilation as well as reflection and rotation. For LG, however, it is easily seen, but nonetheless should be noted, that 2 configurations that are similar up to a dilation cannot be equivalent models of fuzzy indicator data. Thus, to say that 2 LG models are similar (or equivalent) under this test is to say that they are congruent. In general, to claim that two configurations are similar, both the correlation of distances, r_d, and c(X,Y) should be statistically significant as discussed in Borg & Leutner (1985) [7]. For these configurations, with relatively large numbers of points, the two statistics are significant. Thus, it can be safely said that, for analytical purposes, the configurations are essentially identical.

[5] The congruence coefficient is, again, .993, with distance correlation of .959.

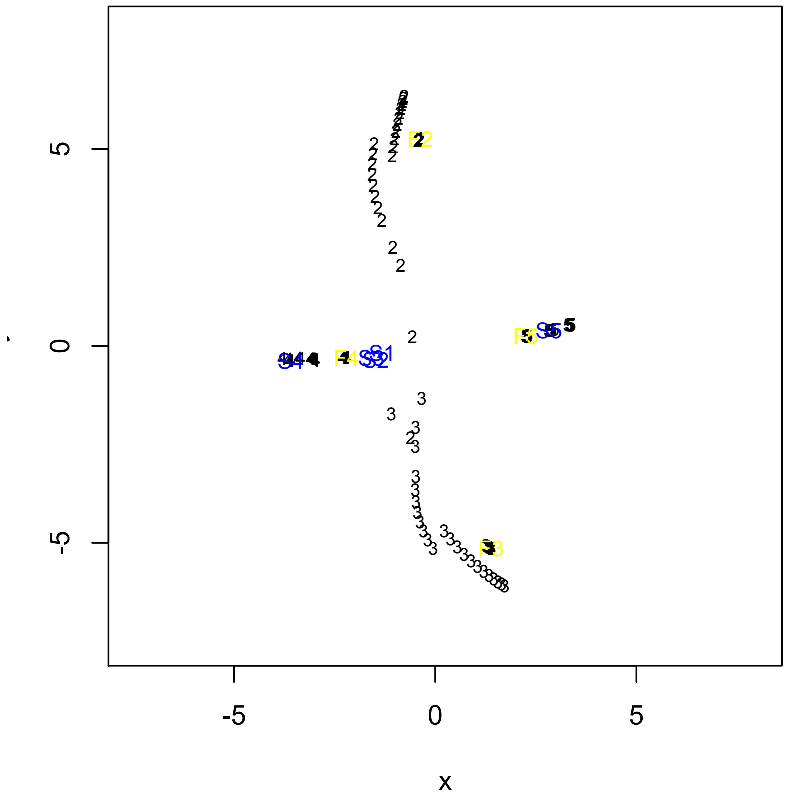

Figure 4.1: 5-State Markov Chain - MPE Model

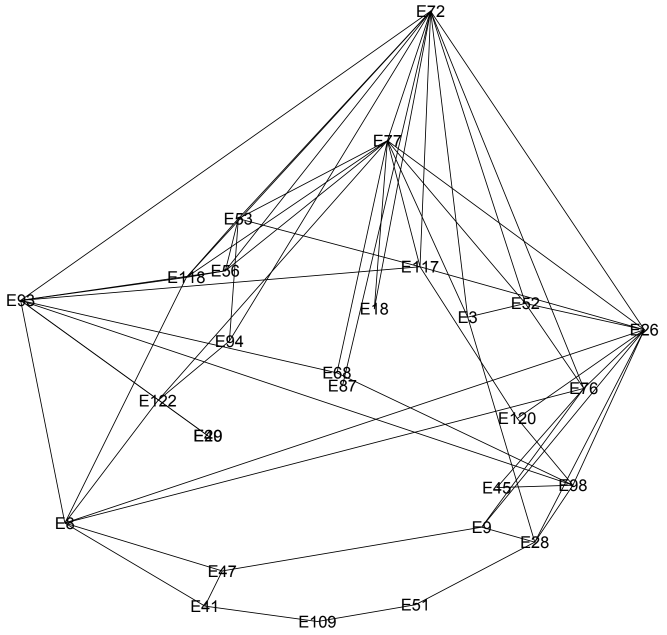

Figure 4.2: French Financial Elite - MPE Model

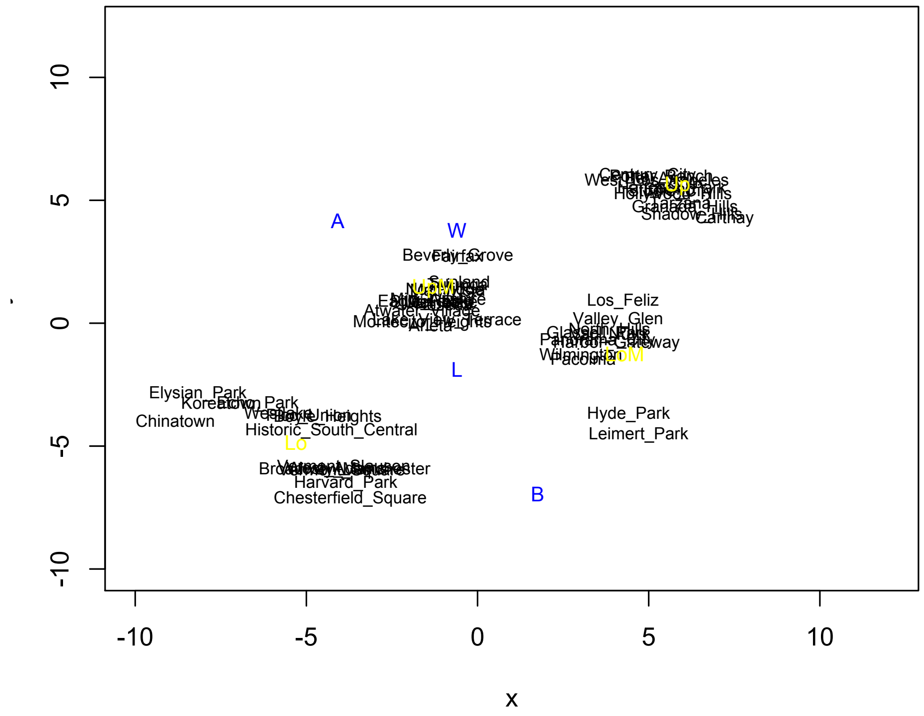

Figure 4.3: LA Nbhds - MPE Model

of the limit configuration suggests itself. We proceed as above to compute a configuration, then use it as the starting point to generate another sequence of vectors for MPE, then repeat this until the desired precision is attained. This is referred to as *cycling*. According to Smith et al., it is reasonable to think, and, in fact, may sometimes be the case, that the new sequence of vectors need only be as long as the number used to compute the limit in the prior iteration. If so, the computing time would increase slightly; however, this is by no means guaranteed.

For the 5-state Markov chain, two cycles, both run by generating a 30-vector sequence, produced a configuration (Figure 4.4) with APWL of .0062, with only an additional 8 minutes of computing time. Again, from an analytical point of view, it clearly resembles the original uLG-1 configuration (with congruence coefficient of .975 and distance correlation of .901).

For the French Financial network, one cycle with a 100-vector sequence produced (with only 2 minutes of additional system time) a very slight improvement in APWL to .0863 and the configuration in Figure 4.5, which again is essentially identical to both the uLG-1 and the MPE configurations (with congruence coefficient of .992 and distance correlation of .954 with the uLG-1 plot). A second cycle of 100 vectors did not result in any improvement.

For the LA Neighborhoods, two cycles, both using the last 26 of 100 vectors, produced a configuration, shown in Figure 4.6, with APWL of .0238. It has congruence coefficient of .996 and distance correlation of .984 with the original uLG-1 plot and is, again, essentially identical.

Finally, we applied MPE with cycling to the constrained LG algorithm using the Swedish Election slide vector model. The constrained LG algorithm uses a fixed-point convergence criterion, so it is not surprising that MPE produces a configuration (Figure 4.7) essentially identical in all respects, though in one-tenth of the time. Cycling did not result in any significant improvement.

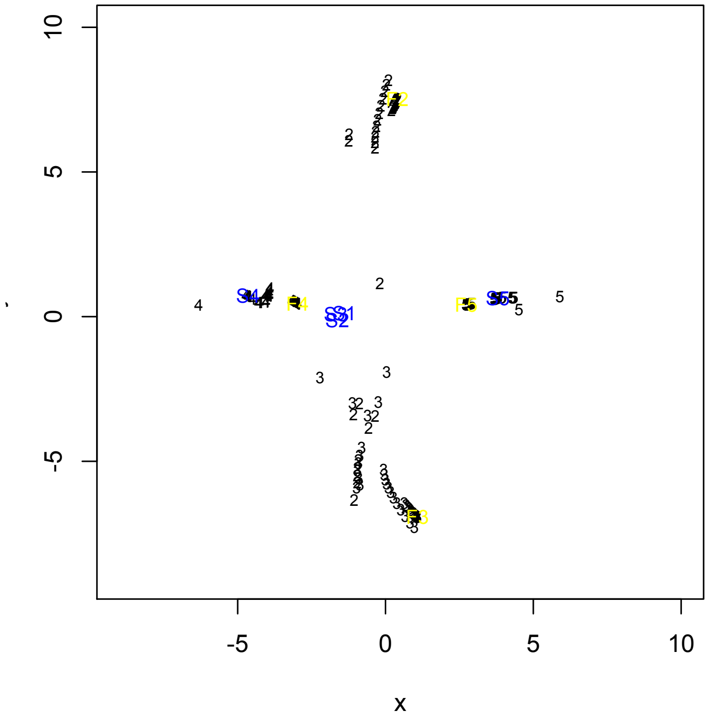

Figure 4.4: 5-State Markov Chain with Cycling

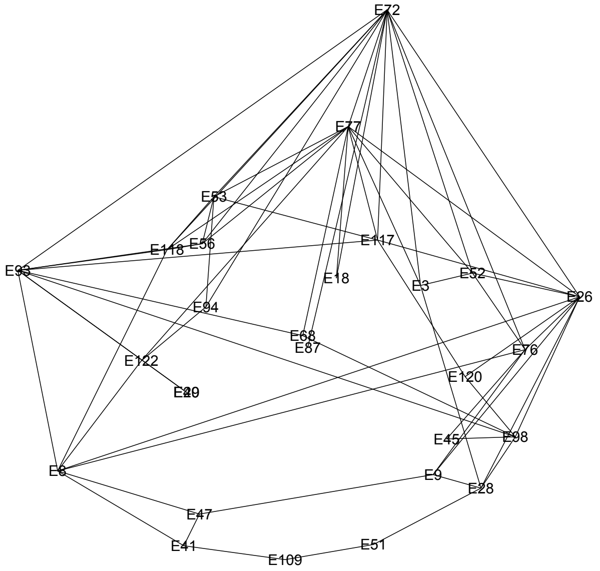

Figure 4.5: French Financial Elite with Cycling

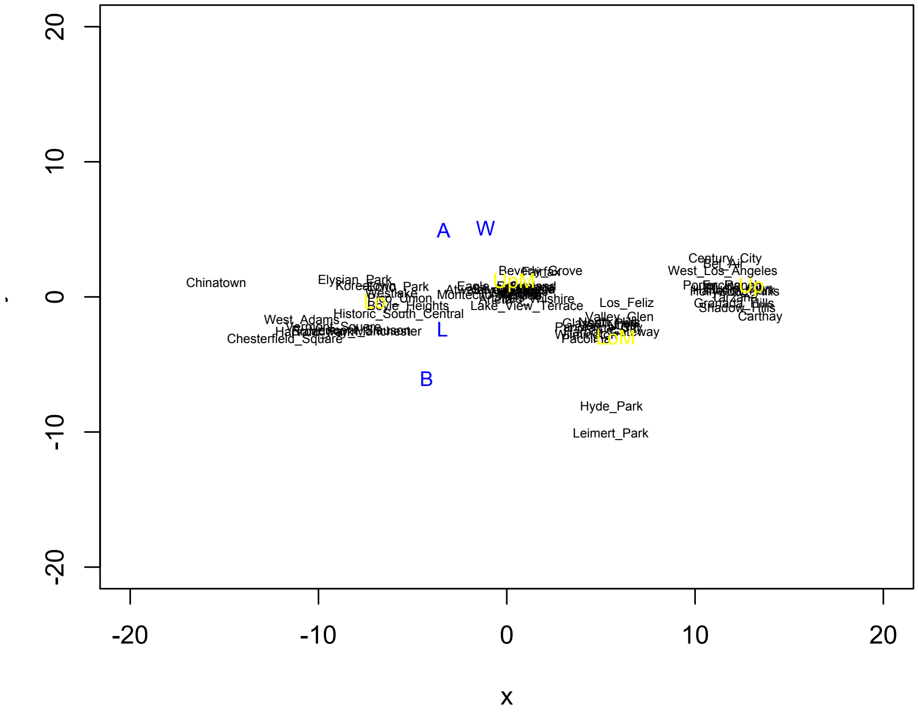

Figure 4.6: LA Nbhds with Cycling

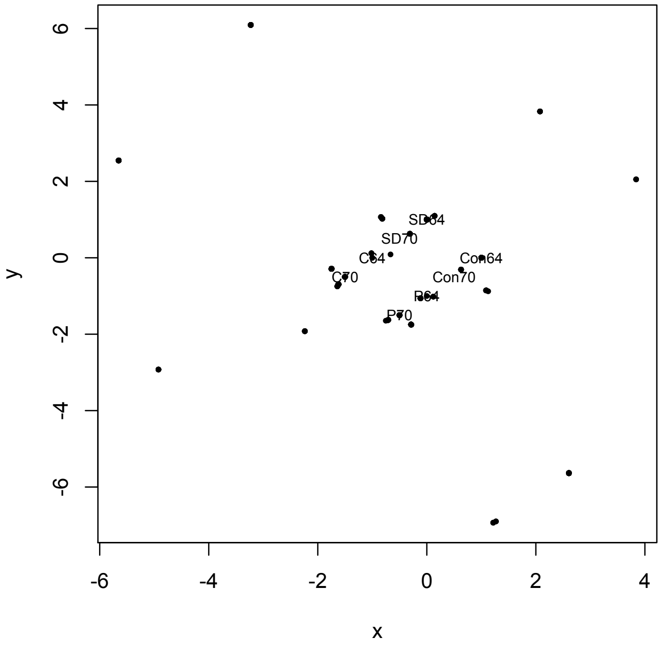

Figure 4.7: Swedish Election Slide-Vector - MPE Model

The conclusion to be drawn from these studies is that MPE provides a very effective method for accelerating the LG algorithm. Considering these empirical results, we can be confident that configurations produced by MPE will be essentially equivalent to those produced by high-iteration runs of the generating algorithm. Key factors in maximizing the acceleration are the number of vectors that must be generated by iterations of LG and the number of cycles that should be run. At this point, there does not seem to be a way to determine the minimum number of vectors needed in advance. On cycling, we follow the recommendation of Smith et al. that cycling should be run until no marked improvement in precision is achieved. From hereon in this research, all uLG-1 models will be run using MPE.

CHAPTER 5

bLG-1: Euclidean LG with Bias Parameters

As noted above, the full LG model of De Leeuw, containing bias parameters is written:

$$\pi_{il}(\text{X,Y}) = \frac{\beta_l exp(-\phi(x_i,y_l))}{\sum_{j=1}^{m}\beta_j exp(-\phi(x_i,y_j))} \tag{48}$$

(for a single indicator matrix). We refer to it as Biased LG[1]. As was briefly discussed previously, it is formally similar to the IPDA of Takane et. al (1987) [63], certain log-linear models and some models from item response theory.

Since IPDA is, in important ways, the most similar method, a review of its development is useful for working with Biased LG. Like LG, IPDA seeks to map subject and category points into a low (usually 2)-dimensional common Euclidean space so that subject points will be closer to the category points they belong to, or are predicted to belong to. This predictive component is the main difference between IPDA and LG, though, as we will see in the next chapter, we can simulate it in LG for categorical predictors.

In IPDA, the probability that object k belongs to group g of a categorical variable with K categories is:

$$\text{p}_{gk} = \frac{\omega_g exp(-\phi_{kg}}{\sum_{h=1}^{K}\omega_h exp(-\phi_{kh})} \tag{49}$$

[1] Again, we mention that this terminology is not intended to connote anything about expectations of parameter estimates

where the ω's are category bias parameters and the ϕ_{ij}'s are squared Euclidean distances between the subject and category (or *ideal*) points.

The model likelihood is:

$$(50) \qquad \prod_k^N \prod_g^K (\mathrm{p}_{gk})^{f_{gk}}$$

where f_{gk} is the category indicator function. Notice the use of squared distances in the probability formula. This allows for optimization of the log-likelihood using Fisher scoring.

It is most important to note here that, in IPDA, the probability in (49) is conditioned on a given vector of predictor values. Thus, the likelihood is conditional as well. That is, if Y is the coordinate matrix of the subject points and X is the matrix of predictor values, we want

$$(51) \qquad \mathrm{Y} = \mathrm{XB}$$

for a matrix of regression coefficients, B. To streamline the IPDA optimization, it is usually assumed that the category points are given by the centroids of the object points belonging to them. Thus, if M is the matrix of category coordinates, with the centroid assumption, we have

$$(52) \qquad \mathrm{M} = (\mathrm{Z}^t\mathrm{Z})^{-1}\mathrm{Z}^t\mathrm{Y} = (\mathrm{Z}^t\mathrm{Z})^-1\mathrm{Z}^t\mathrm{XB}$$

where Z is the indicator matrix of category membership. Therefore, with the centroid assumption, in optimizing the log-likelihood only B and the ω's, the bias parameters, need to be calculated. Biased LG differs in that the predictor constraint is removed allowing for optimization by majorization and MDU methods.

Takane et. al [63] note the evident relationship of IPDA to Coombs's unfolding model, the Luce biased choice model, and to log-linear models (as we have with LG). Another important ancestor, we feel, is Shepard's (1957) [58] model of confusion matrices.[2]

In the case of unconstrained bias parameters, we must compute for a given dataset the configuration coordinates, as is done in uLG-1, and the bias parameters, β's, which give optimum APWL. Notice that, since we are giving a formula for probabilities, the β's must be positive. Also, it is important to note that bias parameters are not uniquely identified. Multiplication of the parameters for a given variable by a positive constant will produce the same APWL since the constant can be factored out of the LG formula. Thus, we can divide each of the parameters for a given variable by their sum and consider them as summing to 1. The algorithm `BiasLG` finds the desired configuration and bias parameters using, as in uLG-1, Euclidean distance. The process, which we call bLG-1, is as follows: First, a configuration is computed using uLG-1. Then bias parameters are computed to minimize APWL starting from that configuration. The bias parameters

[2] Shepard sought to model the probability of stimulus confusions (and associated response confusions) based upon their dissimilarities. To make more rigorous the concept of dissimilarity, Shepard related it to the Euclidean distance between the stimuli in psychological space. He then modeled the probability of a confusion between two stimuli, i and k, as:

$$\mathrm{P}_{i,k} = \frac{W_k exp(-d_{ik})}{\sum_{h=1}^{n} W_h exp(-d_{ih})}.$$

Working from data in the form of symmetric empirical confusion probability matrices, Shepard gave algebraic expressions for the computation of the model distances based upon these probabilities. Weights were introduced into the model to provide for the concept of asymmetric inter-stimuli or inter-response distances; i.e., unfamiliar stimuli are more likely to be confused for familiar ones than vice versa. Algebraic formulas for the weights can be derived from the empirical probabilities, as well. To visualize the dissimilarity relationships, early MDS methods were used to find coordinates for the stimuli and responses (separately) based upon the model distances. Of interest would be to use LG to carry out Shepard's approach. We would have to cast the stimulus-response data in indicator matrix form. Depending on the nature of the data, this could involve fuzzy indicator matrices or, possibly, multiple binary indicators, as we used with the social network and Markov transition data. Then, the LG bias parameters could be constrained to account for asymmetry in the dissimilarities among the stimuli. This is a project for another time, but we mention it here to further show the flexibility and broad applicability of LG.

are computed using the R function `nlminb`, a gradient-based method which uses PORT[3] routines for constrained optimization of general vector functions. Using

$$\beta_l \text{exp}(\text{-}\phi(\text{x}_i,\ \text{y}_l)) = \text{exp}(\text{-}\phi(\text{x}_i,\ \text{y}_l) + \log(\beta_l)), \tag{53}$$

we adjust the configuration distance matrix. The adjusted distance matrix is then used as a new starting configuration for uLG-1, which is run until a desired APWL is reached. New bias parameters are computed for this configuration, and the steps are then repeated until convergence to a desired APWL. The `BiasLG` function requires inputting indicator matrices, a chosen dimension, and maximum iterations for the initial and secondary uLG-1 runs as well as for the bias parameter computation with `nlminb`. As with uLG-1, MPE can be used to accelerate all three steps of the computation. The output gives the initial unbiased configuration as well as the final configuration and the bias parameters and APWL corresponding to both. Any messages from `nlminb` regarding the performance of the optimization algorithm are stored as well. Our primary interest being data visualization, uLG-1 is typically run until a well-fitting configuration is found before computing bias. However, as is suggested by our work in preceding sections, it will no doubt sometimes be the case that bias parameters are sought for fixed or partially fixed configurations and, of course, configurations for fixed bias parameters.

Two questions arise from this algorithmic procedure, one from its construction and the other from its implementation. The first is whether the bias parameter computation process can itself be used to accelerate the computation of configurations with optimal APWL. That is, does the process of adding a small amount to each object-to-category distance, constant in each category, aid in the underlying unfolding process? Unfortunately, but not surprisingly, the answer appears to be

[3] PORT stands for Portable, Outstanding, Reliable, and Tested. The PORT library is a collection of mathematical algorithms, including several for optimization, developed by Bell Labs. For details, see Fox (1997) [29] and Gay (1990) [31]

no. The reason is that the bias distance adjustments are computed to decrease APWL, or, equivalently, to better model the data probabilities. Unlike the MDS additive constants, they are not computed to make distances viable. Thus, we have found that, to find configurations with optimal APWL, the uLG-1 process must do most of the work. The computation of bias parameters and adjustment of distances with them only slightly refines the underlying configurations and only marginally affects APWL. Since these computations are themselves fairly time consuming, the process does not provide acceleration.

The second question is of some importance. It is, what do the bias parameters tell us about our data and the model produced from it; i.e., how should they be interpreted? In the similar types of analyses discussed above, different interpretations are suggested. The bias parameters resemble, for example, item discrimination parameters in IRT. See de Ayala (2009) [11] for further discussion. In log-linear distance modeling of contingency tables, such parameters are generally computed for each dimension and represent either a stretching or shrinking of the dimension. An excellent analysis and interpretation of such parameters is found in De Rooij (2001) [25].

It is clear, though, that neither of these interpretations are applicable to the general LG model. In particular, the bias parameters are not computed for each object or dimension, but separately for each category within each variable. For a particular configuration, they give the one distance (or relative distance) each object must be moved uniformly from or toward each separate category of each variable to give optimum APWL. Geometrically, it may be (and generally is) impossible to carry out or visualize such a move, so these parameters must be thought of in a conceptual way. In IPDA, Takane et al. (1987) [63] suggests that they be viewed as something like a prior group probability or as the marginal effects or overall likelihoods of the columns. This is an interpretation that does seem to apply to LG, at least loosely. Notice that in the absence of any distance parameters, or put another way, with all distances set at 0, all information about the model probabilities is given by the bias parameters. If they were to be computed with distances so constrained, they would give, when put into the LG formula (50), the empirical column marginals. We see that as follows. Consider that, for π_{il} as in (54) (where we are considering the case of 1 variable since bias parameters are computed separately for each variable); that is:

$$(54) \qquad \pi_{il}(\mathrm{X,Y}) = \frac{\beta_l exp(-\phi(x_i,y_l))}{\sum_{j=1}^{m} \beta_j exp(-\phi(x_i,y_j))}$$

the log-likelihood is:

$$(55) \qquad \mathrm{L} = \sum_{i=1}^{N} \sum_{l=1}^{k} g_{il} log \frac{\beta_l exp(-\phi(x_i,y_l))}{\sum_{j=1}^{m} \beta_j exp(-\phi(x_i,y_j))}.$$

Now, suppose object $i \in$ category l. Then $\frac{\partial L}{\partial \beta_l}$ will have the form:

$$(56) \qquad \frac{1}{\beta_l} - \frac{exp(-\phi(x_i,y_l))}{\sum_{j=1}^{m} \beta_j exp(-\phi(x_i,y_j))} = \frac{1}{\beta_l} - \frac{1}{\beta_l}\pi_{il}.$$

Assume there are n$_l$ such terms. Next, suppose $j \in m$, $m \neq l$. Then, $\frac{\partial L}{\partial \beta_l}$ will have n$_m$ terms of the form:

$$(57) \qquad -\frac{exp(-\phi(x_j,y_l))}{\sum_{j=1}^{m} \beta_j exp(-\phi(x_i,y_j))} = -\frac{1}{\beta_l}\pi_{jl}.$$

Continuing in this way, we see that

$$(58) \qquad \frac{\partial L}{\partial \beta_l} = \sum_{i}^{nl} \left(\frac{1}{\beta_l} - \frac{1}{\beta_l}\pi_{il}\right) \text{ - } \sum_{j \notin l} \frac{1}{\beta_l}\pi_{jl}$$

where there are N - n$_l$ terms in the second summation.

Notice that in the absence of any distance parameters, or put another way, with all distances set at 0 (and all information about the model probabilities given by the bias parameters), and keeping in mind the constraint that $\sum \beta_i = 1$, we have:

(59) $$\frac{\partial L}{\partial \beta_l} = \mathrm{n}_l(\frac{1}{\beta_l} \text{ - } 1) \text{ - } (\mathrm{N} \text{ - } \mathrm{n}_l).$$

Setting this equal to 0, we get $\beta_l = \frac{n_l}{N}$, the column (or category) marginals. This justifies our thinking of the bias parameters as prior group probabilities and the applications of constrained bias parameters we will use below.

Conversely, when running unconstrained `BiasLG` on the datasets discussed above, we have found that the bias parameters often come out nearly identical across variables; i.e., they place a uniform multinomial marginal probability across the categories. Thus, the distance parameters contain nearly all information about model probabilities. This is all very much in keeping with what Takane refers to as a prior probability. It also provides some ideas on how to use constraints on the bias parameters for LG modeling.

The most obvious use is to constrain the bias parameters, β_l, to be proportional to category size. This is suggested by Takane as a standard procedure for IPDA, for example. To further investigate the effect of using the bias parameters in this way, we re-ran models using some of the datasets examined previously. First, we used the four-category WHO beverage consumption data. The uLG-1 configuration is shown in Figure 1.2 above. Recall that APWL is .009. Forty-three of the 47 countries (91.5%) are correctly classified by beverage with maximum consumption. To find the bLG-1 configuration, we used the function `ConstrBiasLG`. This function seeks the lowest-APWL LG configuration given a fixed set of bias parameters. The bLG-1 configuration is shown in Figure 5.1, with a detail of the main configuration in Figure 5.2. APWL is somewhat higher at .051, but 44 of the 47 objects (93.6%) are classified correctly. Comparing the two configurations, we notice that, in the bLG-1 plot, the wine and spirits categories are located farther apart allowing somewhat clearer discrimination between countries with these respective preferences. Also, the relative bias for the beer category requires fitting countries with this preference, by far the dominant one, close to the category point, providing for slightly more pronounced clustering of these objects.

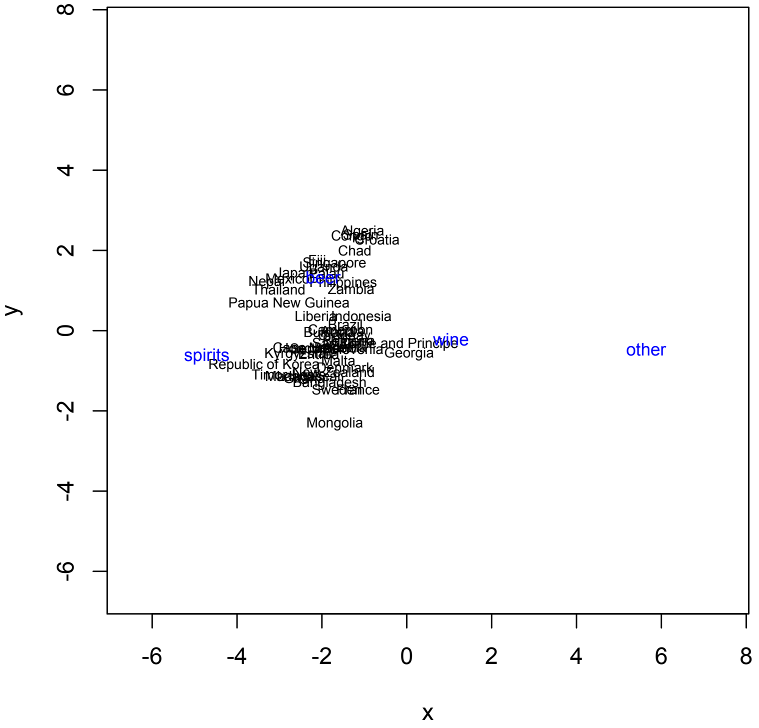

Figure 5.1: WHO Beverage Data with Constrained Bias

Next, we used ConstrBiasLG with fixed bias parameters as discussed above, to create a bLG-1 plot of the LA Neighborhoods data, the uLG-1 plot for which is shown in Figure 22.19 above. For the uLG-1 plot, APWL is .039, with 100%

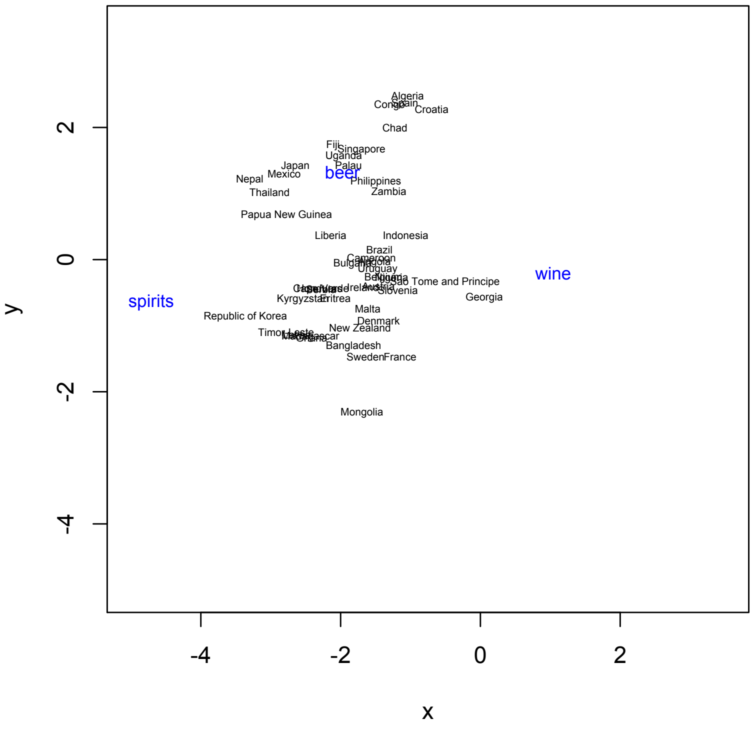

Figure 5.2: WHO Beverage Data with Constrained Bias - Main Configuration

classification of the neighborhoods by income quartile. APWL for the bLG-1 plot (Figure 5.3) is .051, also with 100% classification. Comparing the plots, we notice there is little change in the income quartile category points. This is expected since those categories were essentially unbiased. For the ethnic groups, the White category is moved noticeably toward the center of the configuration, as is the Latino category, though slightly less so. The Asian and Black points are moved slightly farther away. This is the effect we expect and, in fact, that we are seeking. The association of the White category with the upper quartiles and the upward mobility of the Latino group are both effectively highlighted.

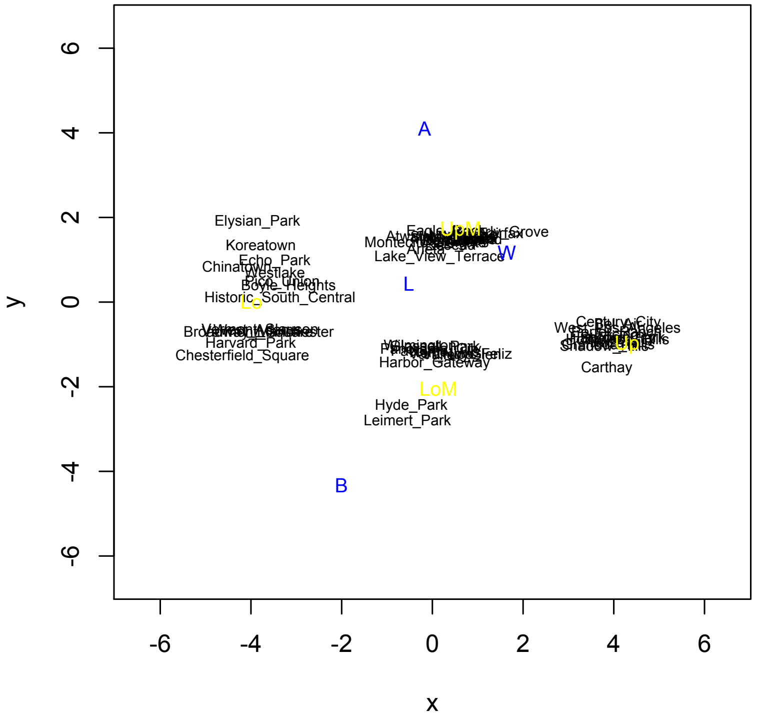

Figure 5.3: LA Nbhds with Constrained Bias

Also of interest with regard to this model are the income quartile profiles produced by the configuration. They are shown in Table 4.1. Comparing them to the uLG-1 profiles, we see that the percentages of Latinos and Whites in the upper quartiles are increased, as are the percentages of Blacks in the lower quartiles while the percentage of Asians in the upper quartile is significantly decreased. The profiles depart slightly more from the mean profiles than the uLG-1 results. This seems to arise directly from the bias of the Asian and Black groups which has led to something of an over-emphasis of certain patterns in the data. By contrast, the Latino and White groups reflect the mean profiles more closely, again with some emphasis of the association between the White group and the upper quartiles.

	Asian	Black	Latino	White
Low	0.081	0.187	0.662	0.070
LoM	0.014	0.291	0.506	0.190
UpM	0.135	0.002	0.316	0.546
Up	0.028	0.020	0.161	0.792

Table 5.1: Income Quartile Ethnic Profiles - Constrained Bias Model

Our most noteworthy results with bias constrained as above come from using `ConstrBiasLG` on the 1-dimensional display of the Swedish Election data. The bLG-1 plot is shown in Figure 5.4 below. Comparing it to the uLG plot, we see some similarities, but also some important differences. As before, there are 4 object points. They are located similarly to Figure 2.36 and they are composed of the same voter groupings. Also, the category (political party) points are positioned in their order on the political spectrum with the large Social Democrat (SD) party positioned well away from the others. Now, however, the SD points are essentially collapsed into one point coincident with the 812 member 3-time SD object group. Again, this is the effect of the bias of the SD categories and it is an emphasizing of an effect we have seen before with uLG, where the algorithm positions large blocks of similarly-profiled objects in this manner as an efficient way to minimize overall APWL. Note also the initial pulling apart of the Center (C) and People's

(P) categories from 1964 to 1968, with the C being pulled toward the object points and the P away from them. This is slightly more pronounced than in the uLG plot. Much more pronounced is the marked shift of the 1970 C, P, and Conservative (Con) points toward the left, with the C point being quite close to the right most object point. These positionings are indicative of the leftward movement of the voters from 1968 to 1970 which manifested itself in significant gains by the Center party. In fact, it gained 95 voters and is the only party to have made overall gains from 1964 to 1970. The bLG plot very effectively captures this feature of the data.

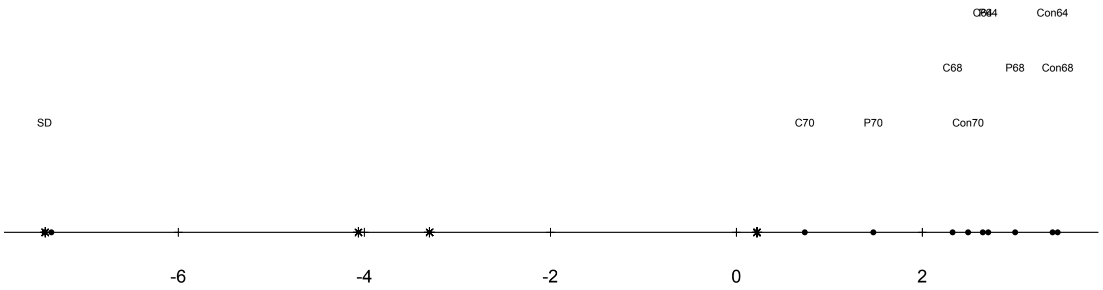

Figure 5.4: Swedish Election Data with Constrained Bias

5.1 Using Bias Constraints for Model Testing

Exploiting the relationships between log-linear models and LG, we can use other bias constraints to perform a type of model testing on certain types of cross-classified data. Consider Table 4.2, taken from Agresti (2010) [2] summarizing ratings of 160 movies by the famous film critics, Siskel and Ebert. Among the questions of interest for this type of matched pair, rater agreement data, is whether the ratings are independent or associated in some way. Between the complete independence and complete agreement, or dependence model (in which all off-diagonal frequencies are 0), there are a number of possible types of association, such as quasi-independence, symmetry, quasi-symmetry (identical to quasi-independence for this 3 × 3 table), diagonals-parameter symmetry, and ordinal agreement, all testable using corresponding log-linear models. The seminal work of Goodman (1979 [34], 1981 [35], & 1983 [36]) elegantly develops many of these ideas. We are interested in studying whether LG with certain bias constraints can be used to explore these possible types of association. We give an example of how such an exploration can be carried out.

	Ebert		
Siskel	Con	Mixed	Pro
Con	24	8	13
Mixed	8	13	11
Pro	10	9	64

Table 5.2: Siskel and Ebert Movie Ratings

To start, we must, as always, transform the data from the tabular form to indicator matrix form. Note that there are two ways to do this. We could construct 2 indicator matrices, one for each rater and each with 3 rating categories, or we could construct a single 9 category indicator matrix, with a category for each of the possible pairs of ratings. Since the relations between these are what we are exploring and what we will be biasing, we take the latter approach. To establish a baseline, a uLG-1 model for this data was run. Because of the exact-clustering property of LG as stated in De Leeuw's IMT, this 9 category, binary data can be fit to almost 0 APWL. Indeed, our model has an APWL of .00014 and 100% object classification accuracy.

We are interested in testing the complete agreement model first. Cursory inspection of the data shows that the raters disagreed on 59 out of 160 movies, so we should not expect this model to fit well. To test this with LG, we place high bias on the diagonal cells and divide the remaining bias among the six other cells. With this bias, the data was rerun in `ConstrBiasLG`. The resulting model has APWL of .040, nearly 277 times the unbiased model, and accurately classifies only 134/160 (83.75%) objects. Because LG is able to nearly perfectly fit the unbiased data, we consider this comparably poor model fit to be a solid indication that the complete agreement model should be rejected. The bias-induced poor LG fit is due to the fact that objects in the low bias cells are able to be positioned further from their category points resulting in some encroachment of objects into incorrect Voronoi cells and there are enough of them to greatly affect the model performance.

We next consider the independence model. This is tested by placing high bias on the off-diagonal cells and the remaining bias on the diagonals. For this model, APWL is .0197, around 140 times the unbiased model, and classification accuracy is 91.875% (147/160). Again, this is fairly poor considering the nature of this data. Based on these results, we are inclined to reject both of these models, as did Agresti in his log-linear analysis of the data.

As noted above, between these extremes there are generally several possibilities for association. With a table of these dimensions, these are subsumed into essentially two possibilities, the quasi-independence (QI) and the ordinal (and reverse-ordinal) agreement models. We first consider the ordinal agreement models. Ordinal agreement postulates that, given rater disagreement, there is a tendency for higher ratings by one rater to occur with relatively high ratings by the other. Reverse-ordinal agreement is, of course, the converse. Formally, the models are given by

$$\log \mu_{ij} = \lambda + \lambda_i{}^A + \lambda_j{}^B + \beta \mathrm{u}_i \mathrm{u}_j + \delta \mathrm{I}(\mathrm{i} = \mathrm{j}) \tag{60}$$

where the parameters β and δ added to the independence model describe association off the diagonal and beyond-chance agreement on the diagonal respectively. The u's are ordered category scores. The sign of β determines whether ordinal or reverse-ordinal agreement exists. Without the β term, we have the formula for the QI model.

To use LG to test ordinal agreement on this table is straightforward. We place low bias on cells (1,3) and (3,1) of the table, those where Siskel's lowest rating is matched with Ebert's highest and vice versa, the only cells where large frequencies would not be consistent with the null hypothesis. `ConstrBiasLG` yields a model with APWL of .0234 and correct classification of only 150 of 160 objects (93.75%). For the reverse ordinal agreement, we adjust the bias of cells (1,2) and (3,2), which yields APWL of .0205 and correct classification of 143 of 160 objects (89.375%). Based on these results, we reject the ordinal agreement models, as does the log-linear approach (which finds the β parameter to be insignificant).

This leaves the QI model. For quasi-independence, given rater disagreement, the ratings are independent. Constructing a bias scheme for this is difficult. Following our approach above, we'll want to set an equal bias for the diagonal cells with the off-diagonal cells equally dividing the remaining bias. What those biases are to be, however, cannot be determined *a priori*. Notice, for example, that both the complete agreement and independence models are examples of this type of biasing scheme. In fact, they trivially satisfy the quasi-independence definition and can be taken as extreme examples of the QI model. This suggests a method for testing QI. Over 200 iterations of bias, we uniformly vary the biases from the complete agreement setting to the independence setting and measure the APWL's resulting from the `ConstrBiasLG` configurations. The left-hand side of Figure 5.5 below shows these plotted by iteration. Notice that, only for the extreme bias settings; i.e., those closest to the complete agreement or independence settings, do we see significant increases in APWL. Classification accuracy decreases only at even slightly more extreme cases. This stability of the configurations under these bias schemes suggests that quasi-independence settings fit the data well. For comparison, the right-hand figure gives the same plot as the ordinal agreement settings are varied uniformly between ordinal and reverse-ordinal. We observe exactly the opposite effect. The ordinal agreement models result in poor-fitting configurations. For the poorest fitting model, classification drops to 75% (120/160) which is strikingly weak performance for LG on this type of data.

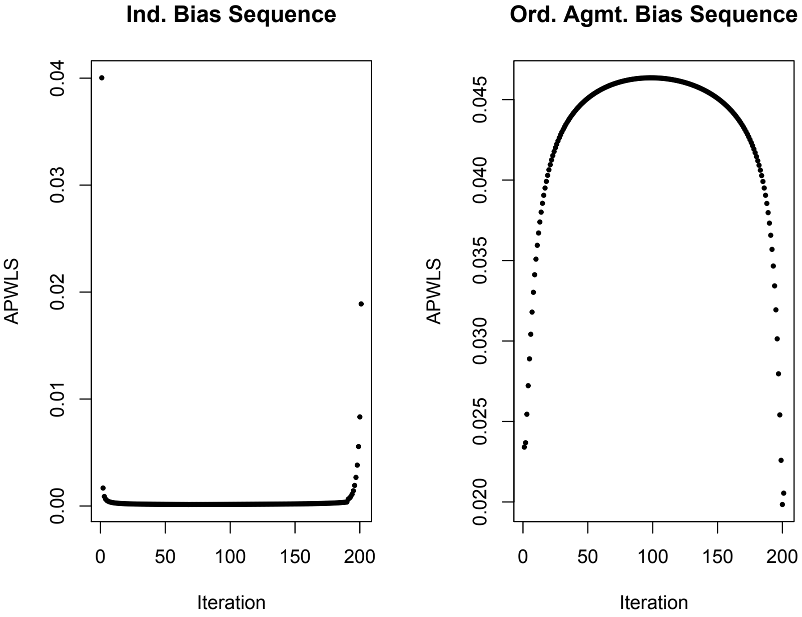

Figure 5.5: APWL Patterns for QI and Ord. Agmt. Bias Schemes

To further examine the soundness of this technique, we apply it to the synthetic data in Table 4.3 for which the ordinal agreement model is clearly appropriate. The unbiased model has APWL of .00013. Biasing for ordinal agreement gives an APWL of .0012, about 9 times that of the overfitted model, with 100% classification accuracy. The reverse ordinal agreement biasing with high probability of overall agreement gives APWL of .0039, about 30 times that of the baseline model.

Biasing for reverse ordinal agreement with low probability of overall agreement gives APWL of .0083, 63 times the baseline. Finally, the corresponding independence models give APWL's of .0132 and .0173, 102 and 133 times the baseline, respectively. Interestingly, classification accuracy for all of these biased models is still 100%. This may be due to the fact that, for a table of such small dimensions and with a small sample size, it is difficult for a biasing scheme to distinguish between these types of associations. Despite this, the APWL results are quite in line with what this method should demonstrate and suggest that, with further refinement, it can be highly effective for testing various contingency table associations.

	Rater 1		
Rater 2	Con	Mixed	Pro
Con	24	20	2
Mixed	20	13	20
Pro	2	20	64

Table 5.3: Synthetic Rater Agreement Data

5.2 Bias Constraints and Visualizing Interaction Effects

A final application we will consider of bias constraining is visualization of interactions among predictors in logit-based regression. As we have seen from our regression studies, LG allows us to visualize the relationship among predictors, but, using our previous approach, it is relatively difficult to assess interactions. This is because it is difficult to precisely locate the ideal point for objects with each of the interaction profiles. To overcome this, we can use a combination of fuzzy indicators and bias constraints. Again, we use an example to illustrate the method.

Consider the data below (Table 5.4) on mental impairment (MI) by socioeconomic status (SES) and life events (LE) taken, again, from Agresti (2010) [2]. In Agresti's regression analysis, the predictors are SES, a binary variable (0 for low, 1 for high), and LE, a composite measure of the number and severity of recent stressful life events undergone by the subject, recorded as an ordinal variable ranging from 0 to 9. The response is MI, a 4 category ordinal variable. We are interested in visualizing the interaction, if any, between SES and LE and the 0, 4, and 9 levels, these being the low, median, and high levels for the LE variable.

Some difficulties are immediately apparent. We have only very few subjects with the six combined predictor profiles we are interested in. And these have different MI scores which means that, using binary indicators, we would not produce a single point for such objects. To overcome these obstacles, we first create a larger population using bootstrap-type resampling, of size 200 for this instructional example. Next, we identify all of the objects with the 6 predictor profiles we are interested in and compute a fuzzy (i.e., probabilistic) indicator matrix for their probability scores. Notice that this leaves us with only 6 objects in our dataset and, even with a fuzzy indicator matrix, we are not likely to get interesting or representative results using `UnbiasLG`. However, we will bias the SES and LE variables based on their sample proportions to better reflect the population.

Using the approach, `ConstrBiasLG` produces the plot shown in Figure 5.6. We submit that, from this plot, we can readily understand both the relationships between the predictors and the response variable and the interactions among the predictors at these levels. Certainly, it is clear that high SES is associated with low levels of impairment and vice versa. For LE, the association from low to high is nowhere near as stark. Regarding interactions, notice that high SES makes relative lack of impairment much higher at 0 LE, slightly higher at 4 LE, and has essentially no effect at 9 LE. On the other hand, for a subject with low SES, 0 LE is nearly indistinguishable from a 4 LE and, in fact, substantially more likely to

Subj	MI	SES	LE	Subj	MI	SES	LE
1	W	1	1	21	Mild	1	9
2	W	1	9	22	Mild	0	3
3	W	1	4	23	Mild	1	3
4	W	1	3	24	Mild	1	1
5	W	0	2	25	Mod	0	0
6	W	1	0	26	Mod	1	4
7	W	0	1	27	Mod	0	3
8	W	1	3	28	Mod	0	9
9	W	1	3	29	Mod	1	6
10	W	1	7	30	Mod	0	4
11	W	0	1	31	Mod	0	3
12	W	0	2	32	Imp	1	8
13	Mild	1	5	33	Imp	1	2
14	Mild	0	6	34	Imp	1	7
15	Mild	1	3	35	Imp	0	5
16	Mild	0	1	36	Imp	0	4
17	Mild	1	8	37	Imp	0	4
18	Mild	1	2	38	Imp	1	8
19	Mild	0	5	39	Imp	0	8
20	Mild	1	5	40	Imp	0	9

Table 5.4: Mental Impairment by SES and Life Events

be impaired than a 4 LE with high SES. For 9 on LE, low SES accentuates the tendency toward impairment. These observations are consistent with Agresti's regression results.

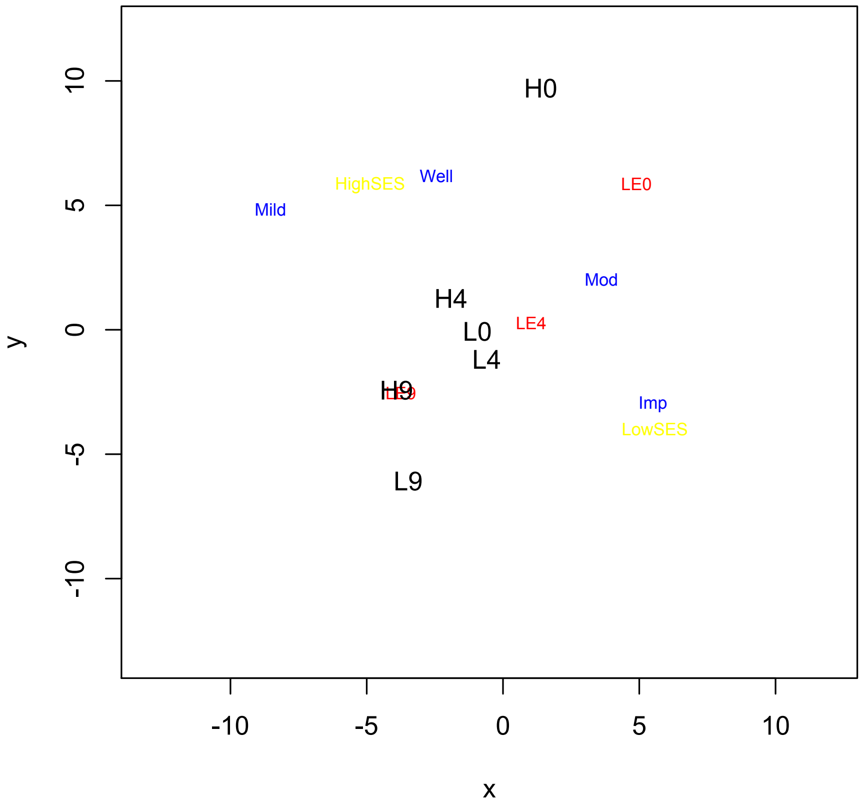

Figure 5.6: Mental Impairment Predictor Interactions

CHAPTER 6

LG and IPDA

At this point, it will be worthwhile to further explore the similarities between LG and Takane's IPDA which we've noted above. In particular, now that we have closely examined LG, with and without bias, we are in a position to compare the configurations obtained by De Leeuw's and Takane's methods. We start by modeling Guilford's (1936) [38] data on judgment of lifted weights, shown in Table 6.1. It is taken from Takane (1989) [62], where it was analyzed by IPDA. The reader is referred to the Takane paper where his results, including a 2-dimensional configuration, are given.

Weight(gms)		Judgment: A is			
A	B	Greater	Doubtful	Less	Total
185	200	5	4	91	100
190	200	12	18	70	100
195	200	15	25	60	100
200	200	30	42	28	100
205	200	55	35	10	100
210	200	70	18	12	100
215	200	85	9	6	100
Total		272	151	277	700

Table 6.1: Guilford Weight Judgment Data

To model this data in LG, we first convert it to indicator matrix form, using two indicator matrices, each with 700 object rows. The first matrix has 7 columns, one for each of the weighting schemes, while the second has three columns corresponding to the three possible judgments. Notice that biasing here is unnecessary since there are the same number of objects in each of 7 object classes. IPDA produces essentially a one-dimensional solution for this data, but the weight-pairing and judgment points lie on a quadratic. From psychometric theory, this is viewed as an important aspect of this data. `UnbiasLG` produces the 2-dimensional plot shown in Figure 6.1. Like the IPDA configuration, the weight pairs are arrayed in essentially a quadratic pattern. In contrast to the IPDA results, the judgment points are not placed along that quadratic. Instead, they are moved to the interior of the curve and placed in strata to correspond to the weight pairs that tend to produce the judgments. We have seen this type of stratification before in LG plots and it appears to be a common characteristic of the clustering method. It seems to be very effective here in capturing the relationships in the data. Clearly, a judgment of `Heavier` (the large 3 in the plot) is closely associated with pairings 6 and 7, to which it most directly corresponds, and slightly less so with pairing 5 (not quite as heavy). Also, a judgment of `Lighter` (the large 1) is associated with pairings 1, 2, and 3. For this plot, notice that a judgment of `Doubtful` (the large 2) is closely associated with pairing 4 (equal weights), as is expected, but associated more closely to pairing 5 (A is heavier) than 3 (A is lighter). In the IPDA plot, the opposite is true. Here, LG seems to reflect the data slightly more accurately since 35 doubtful responses come from pairing 5 while only 25 come from pairing 3.

Our second example involves Maxwell's (1961) [47] hypothetical mental health data, taken from Takane (1987) [61] and analyzed by IPDA, therein. It is shown in Table 6.2. The predictor patterns give presence or absence of the following symptoms: anxiety, suspicion, thought disorder, and guilt, then the subjects are diagnosed as schizophrenic, manic-depressive, or anxiety disordered .

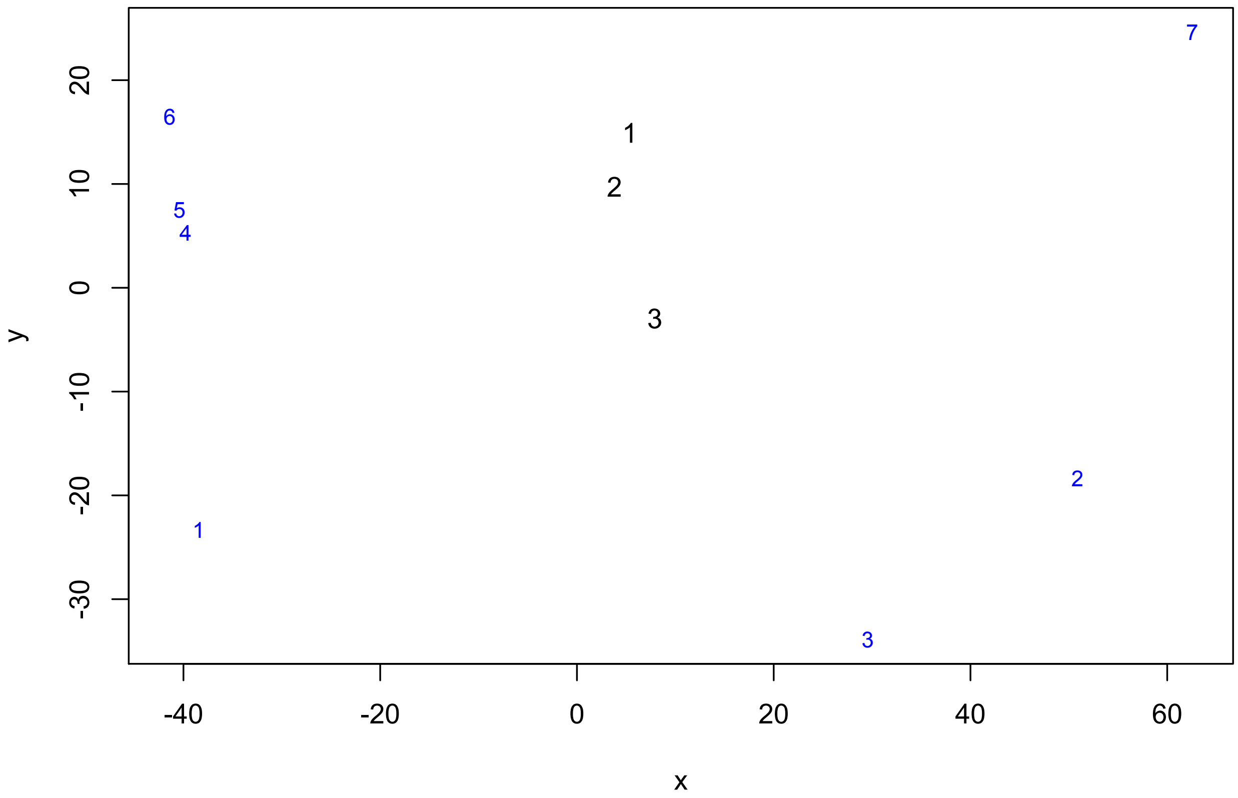

Figure 6.1: Guilford Weight Judgment Data

Profile					Frequencies		
	A	S	T	G	SC	MD	AX
1	0	0	0	0	38	69	6
2	0	0	0	1	4	36	0
3	0	0	1	0	29	0	0
4	0	0	1	1	9	0	0
5	0	1	0	0	22	8	1
6	0	1	0	1	5	9	0
7	0	1	1	0	35	0	0
8	0	1	1	1	8	2	0
9	1	0	0	0	14	80	92
10	1	0	0	1	3	45	3
11	1	0	1	0	11	1	0
12	1	0	1	1	2	2	0
13	1	1	0	0	9	10	14
14	1	1	0	1	6	16	1
15	1	1	1	0	19	0	0
16	1	1	1	1	10	1	0
Total					224	279	117

Table 6.2: Maxwell Mental Health Data

For LG analysis, there are two possible approaches. First, we can deconstruct the data, as we've done before, giving us five indicator matrices, each with 620 rows (one for each patient). There will be one matrix for each of the symptoms (which will each have 2 columns for absence or presence of the symptom) and one three column matrix classifying the patient by diagnosis. `mpeLG1` produces the configuration shown on the left of Figure 6.2. This plot provides some important information about the data. The clearest association is between thought disorder and schizophrenia. Schizophrenia is also most marked by presence of suspicion. Anxiety disorder is, of course, most closely associated with presence of anxiety and, somewhat less, with absence of guilt. Manic-depression (MD) is also marked by presence of anxiety and, in fact, has a similar symptom profile to anxiety disorder, but for the much closer association of MD with presence of guilt. IPDA does not easily produce such a plot. This is because it views the 16 predictor patterns as its objects and views the predictors themselves as explanatory rather than as variables to be jointly scaled and plotted as we have done here with LG.

This brings us to the second LG approach, which more closely follows IPDA. Here, we cast the data as two indicator matrices, each with 16 object rows. The objects are the predictor patterns. The first matrix is a 16 × 16 identity matrix; i.e., its categories represent predictor patterns, the same as its objects. The second is a three column fuzzy indicator matrix in which the frequencies shown in the table are converted to probabilities. In this approach, biasing by column membership is important since the predictor groups are different sizes, as are the diagnosis groups. With this bias, `ConstrBiasLG` produces the configuration on the right of Figure 6.2. This is similar to the plots produced by IPDA, except that, again, LG produces the stratification effect in the diagnosis points. The plot on the left is helpful to understand and interpret the right-side plot. Notice that the predictor profiles in the SC (schizophrenia) strata are all those in which thought disorder is positive, except profile 5 which is positive for suspicion alone. Profile 9,

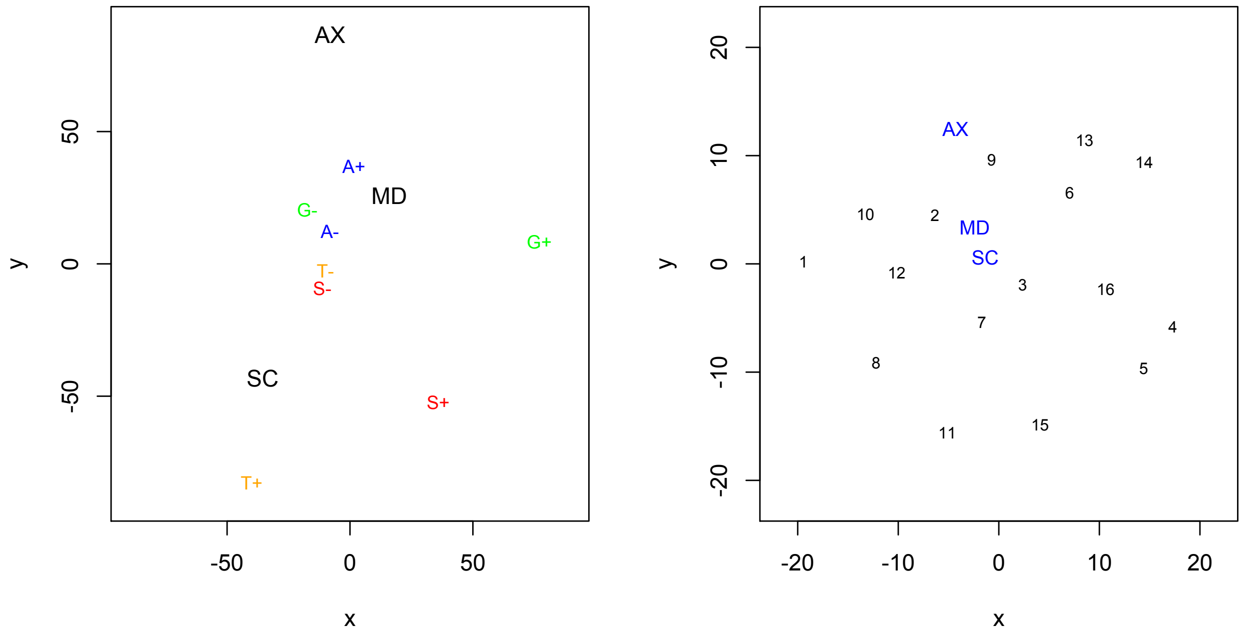

Figure 6.2: Maxwell Data - Two LG Approaches

positive on anxiety alone, is most closely associated with anxiety disorder, while profile 2, guilt alone, is with manic-depression. Other interesting associations can be identified as well. The reader is referred to Takane's (1987) [61] results for further comparison.

CHAPTER 7

Stability Studies

We have presented LG as primarily an exploratory and visualization method. Nonetheless, as with any such method, it is important to consider whether the patterns and relationships we remark upon in the created plots are meaningful or whether they are chance occurrences elevated to meaning by what has been referred to as *magical* thinking. (See, for example, Diaconis (1985) [27]). This raises questions of *stability*, a central concept in the Gifi system. (For an excellent discussion of this, see Michailidis and De Leeuw (1998) [49].)

We say that data analysis results are stable if small, unimportant changes in the analytical input (for LG, the indictor matrices) lead to negligible changes in the output (the configuration). There are two types of stability to consider, *internal* and *external.* We begin with internal stability, which is the more relevant to exploratory methods. It refers to a plot's resistance to outliers and other potential influential features of a dataset. It gives a sense of how well the plot summarizes the data at hand. An important objective of exploratory data analysis being to thoroughly know one's data before moving to modeling and questions of inference, internal stability is an essential feature of any exploratory method. It can be thought of as a type of robustness.

One of the most important forms of internal stability is known as *data selection stability.* A model or method has data selection stability if variations in the input data, such as omitting outlier objects or collapsing categories, do not cause unexpected variations or breakdowns in the output. To provide an example of an internal stability test in LG, we consider the data selection stability of our first model of the Maxwell data, plotted on the left of Figure 6.2. We will consider this to be the baseline output for the test. Recall that this was constructed using data in the form of 620 objects with five binary indicator matrices giving the status of the four symptom diagnoses and the final disease diagnosis. First, notice that the data has several final diagnosis categories that contain only one or two objects. A standard stability test is to remove these to see if such unusual observations unduly affect the output. Having done this and computed a new category plot, we need a way to compare this to the baseline. This can be done fairly well, in this case, just by visual inspection, though with any much larger plot that would be impracticable. The plot is shown in Figure 7.1 and is strikingly similar to the baseline plot. There are only slight differences in category positioning that do not alter our assessment of the relationships between the variables. Even noting this, however, since we are interested in the precise distance relationships between the categories, some finer measures need to be evaluated. We shall use the congruence coefficient and distance correlations, which compare these distance relationships in the aggregate, as well as APWL and classification accuracy, which give some measure of the object alignment. In this case, all of these are consistent with our visual comparison. The congruence coefficient with the baseline is .989 with distance correlation of .944. The APWL of the test plot is .038 compared to .045 for the baseline with classification rates of .997 versus .989.[1]

Another type of data selection stability test involves collapsing of categories.

[1] It is worth considering these results, which are fairly common for LG used on moderate-sized data sets, in light of the Ideal Model Theorem. As we have noted above, if a plotting technique, such as `homals`, produces a configuration with 100% classification, sufficient dilation of the configuration will produce a plot with arbitrarily small APWL under the LG metric. Now suppose we have a configuration with *nearly* 100% classification produced by LG. From the study we have done here, we see that it is likely that replotting after removal of the few misclassified objects would not significantly change the overall configuration. Thus, after removing these objects from the plot (giving 100% classification), we can dilate to achieve an arbitrarily low-APWL LG configuration of the remaining data which we can be reasonably certain gives a fair picture of the complete dataset. This can be a useful time-saving approach if, for example, the near-100% plot was produced after relatively few iterations of the algorithm.

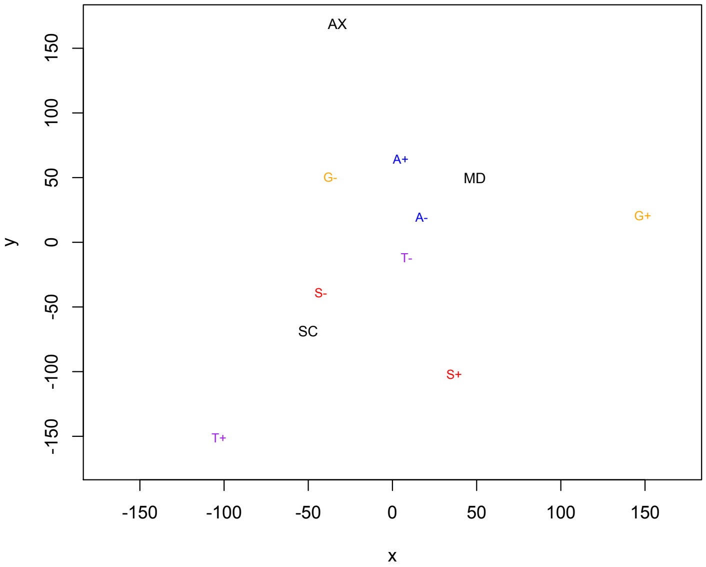

Figure 7.1: Maxwell Data - Outlier Objects Removed

In the Maxwell data, for example, a researcher might notice that 110 of the 117 anxiety disorder diagnoses are positive for the anxiety symptom, with six of the remaining seven being recorded as asymptomatic. Ninety-two of the 110 are positive on anxiety alone. This might suggest that the other symptoms evinced by these patients are irrelevant or possibly even erroneous. If this is the case and we collapse these 117 objects into just one profile category - positive on anxiety alone - a robust visualization of the data should not be greatly affected. We see that here for the LG method. The congruence coefficient with the baseline is .967 with .889 distance correlation. APWL is .032 with classification accuracy of .988. An inspection of the plot, shown in Figure 7.2, again shows only slight alteration in category positions and none that substantially affect our sense of the variable relationships.

Another test of this type on this data involves the schizophrenia diagnosis. We notice that profiles 3, 4, 7, and 15 include a total of 92 patients, all of whom

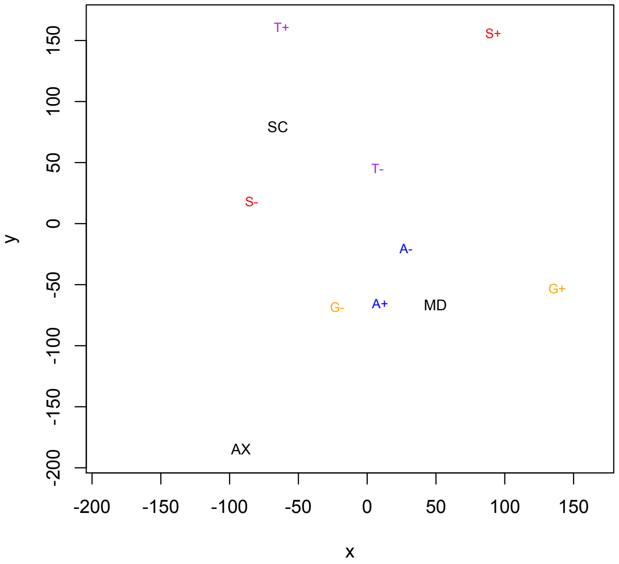

Figure 7.2: Maxwell Data - Anxiety Diagnosis Collapsed

are diagnosed as schizophrenic. That all of these objects have the same diagnosis suggests that these profiles should be identically positioned in the model space. The common characteristic of these profiles being a positive thought disorder finding, we collapse all 92 of these objects into that symptom profile. We obtain a congruence coefficient of .995 with the baseline model with distance correlation of .975. APWL is .033 with classification accuracy of .995. The plot is shown in Figure 7.3. As expected given these readings, it is quite difficult to detect any significant differences between this plot and the baseline plot. Interestingly, these last two configurations are seen to be highly similar. In fact, they have congruence coefficient of .976 and distance correlation of .919. These results suggest that LG is indeed capturing real structure in the data as opposed to easily-varied chance features. Of course, these are results for just one data set, but our experiments with LG suggest that this type of internal stability is characteristic of the method.

A logical next step is to combine these collapsings. It should be kept in mind

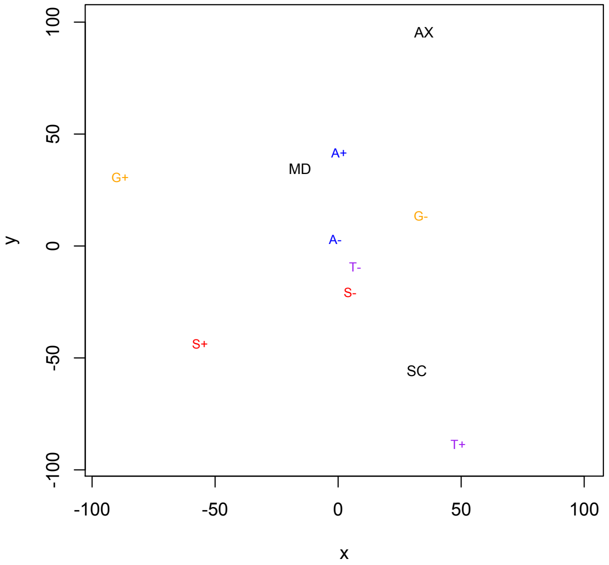

Figure 7.3: Maxwell Data - Schizophrenia Diagnosis Collapsed

that this involves altering the profiles of 88 of the 620 objects in the data set. The resulting plot is shown in Figure 7.4. Its congruence coefficient with the baseline is .943, which, as we will see, is actually a borderline result. The distance correlation is .714, which indicates some clear difference in positioning among these plots. Inspection of Figure 7.4 reveals some of these. The diagnoses categories are fairly similarly aligned, but, not surprisingly, the Anxiety and Thought Disorder symptom categories are positioned differently, reflecting the sharpened identification of these symptoms with the Anxiety and Schizophrenia diagnoses, respectively. Notice that, even with the altered positioning, the relationships between the symptom and diagnoses points we would draw from this plot are essentially the same as those we see in the baseline.

We turn next to external stability. A method is externally stable if samples from the same population give essentially the same output. This is also known as *replication stability.* Thus, external stability typically involves measures of statis-

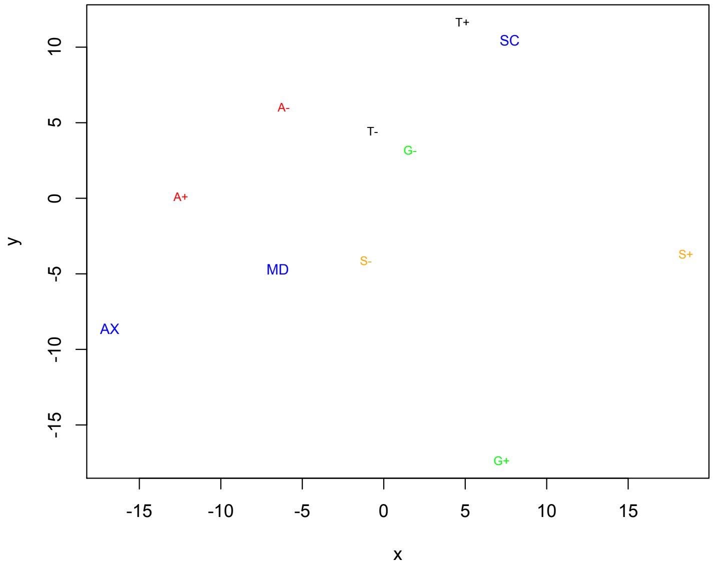

Figure 7.4: Maxwell Data - Anxiety and Schizophrenia Diagnoses Collapsed

tical significance and confidence levels in inferential and confirmatory analysis.

Although LG is a primarily an exploratory method, we wish to have some way to assess if and to what extent our observations about the input variables can be generalized to some larger population. We will attempt this, again, with the Maxwell data. In this test, we will use the second of the LG approaches, plotting the 16 symptom profiles using bias parameters. The bias parameters provide a way of accounting to some degree for sampling variability. Samples are obtained using the bootstrap. Following the approach of Michailidis and De Leeuw (1998) [49] for using the bootstrap in homogeneity analysis, we sample objects (or data rows) with replacement using 100 samples of the same initial population size of 620. The data are then tabulated, as in Table 6.2, then profiles are computed using a fuzzy indicator and constrained bias parameters, as with the right hand plot in Figure 6.2.

Evaluating the replication stability of these results is not as straightforward as with classical point processes. We first consider confidence regions for the object and category points, but we should note that since we are mainly interested in distance relationships among these points, the regions can be quite difficult to evaluate. As an example, the main diagnosis configurations for our 100 runs are plotted in the Figure 7.5 (after being centered and oriented). It shows a tendency for these plots to preserve the strata relationships among the categories. We also see that the manic-depression point has the smallest variation and anxiety the largest. This is how we want the method to behave, but beyond this, it is difficult to draw any precise conclusions.

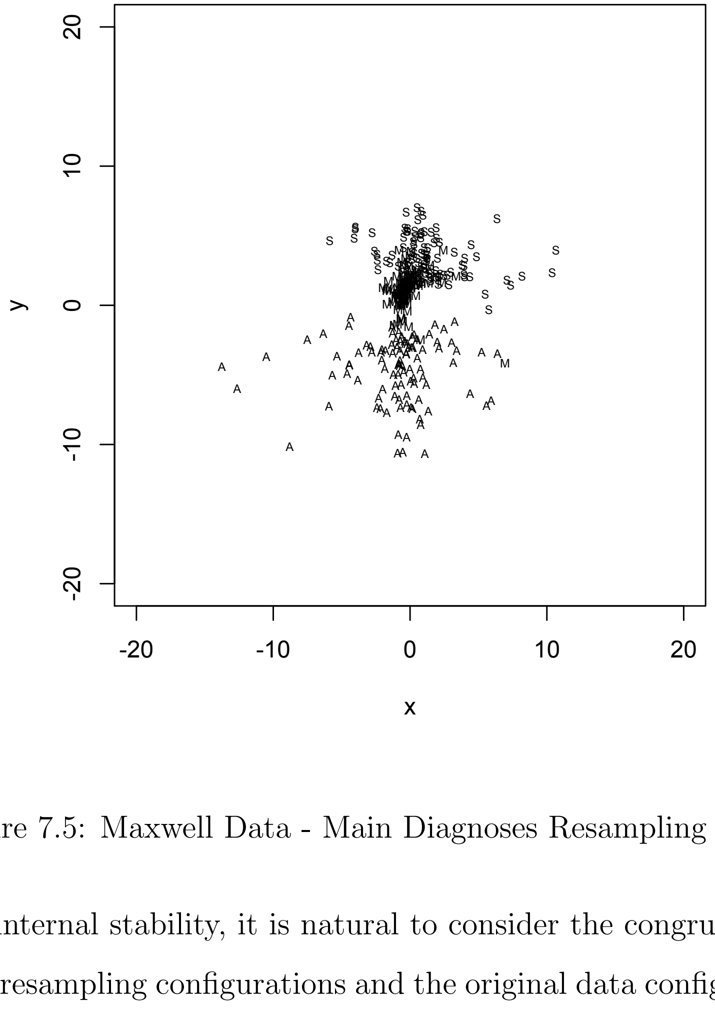

Figure 7.5: Maxwell Data - Main Diagnoses Resampling Regions

As with internal stability, it is natural to consider the congruence coefficients between the resampling configurations and the original data configuration. Figure 7.6 gives their distribution. At first, it is encouraging to note that only one of these is below .900. But for this configuration, the distance correlation is .435 and the configuration is actually quite different from the baseline. This is true, as well, of the other 6 outlier configurations. Further, for the low outlier, only 7 of the profiles give the same maximum probability for the main diagnoses, only 9 for the high outlier. Thus, some further examination is required. For the low, non-outlier, the configuration is plotted in Figure 7.7. It has congruence coefficient of .929 and distance correlation of .816. We can see that it much more closely resembles the baseline. Although profiles 6, 8, 10, and 13 are quite differently plotted, it is fair to say that we get a picture of the data roughly consistent with the baseline. It turns out that the congruence coefficients and the distance correlations have a correlation of .996. Thus, as we move higher in congruence coefficient, we obtain configurations that more and more closely resemble the baseline. Figure 7.8 shows the probability residuals for the main diagnoses profiles for the configurations at the 1st quartile congruence coefficient (.948, red), median (.959, orange), 3rd quartile (.967, blue), and maximum (.989, black). The dotted lines are at ±.2. Note that the higher residuals appear at profiles 5 and 6, 8, and 12 and 13. These are profiles with relatively mixed main diagnoses patterns. We also note that, from the 1st quartile on, these configurations appear to be relatively stable in terms of modeling the main diagnoses probabilities; that is to say, relatively stable in terms of profile positioning. To summarize, in testing LG models for replication stability, the congruence coefficient can be used, but it must be kept in mind that coefficients much below .95 must be studied further to see how they vary from one's baseline model. Examining the configurations themselves, computing distance correlations, and plotting residuals can all be done to determine if replication stability exists.

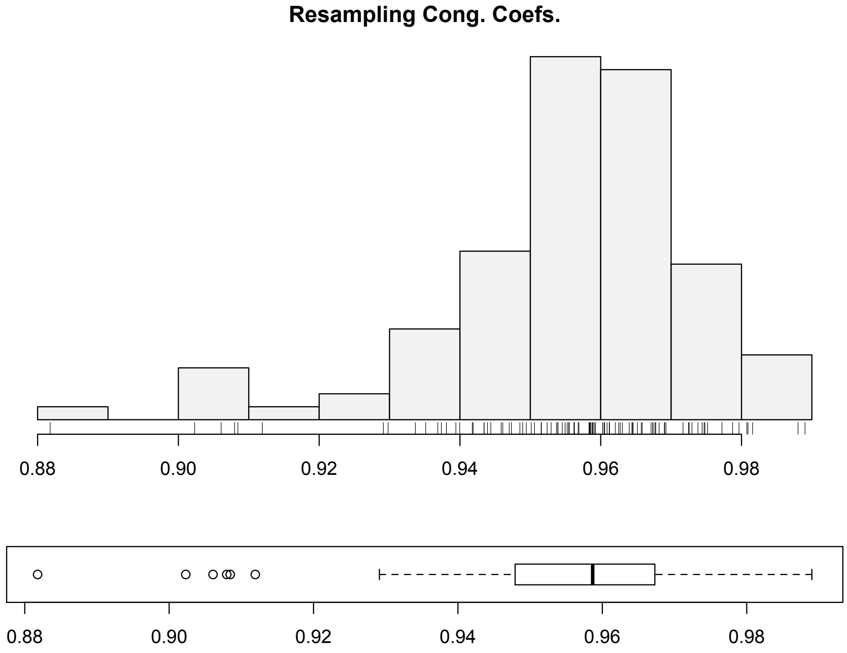

Figure 7.6: Maxwell Data - Distribution of Congruence Coefficients

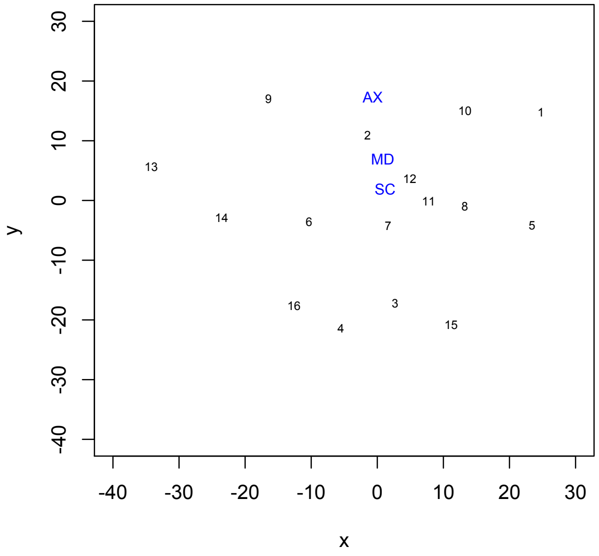

Figure 7.7: Maxwell Data - Low Non-Outlier Configuration

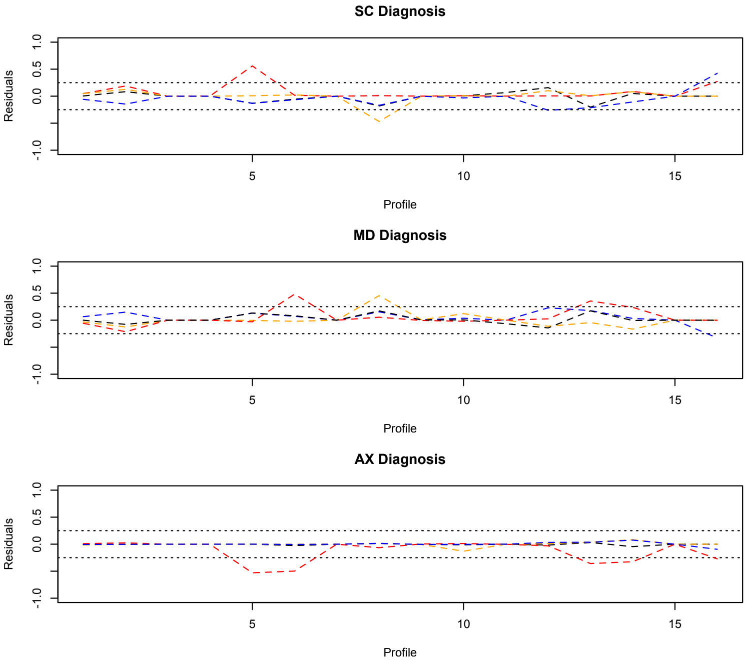

Figure 7.8: Maxwell Data - Resampling Residuals

CHAPTER 8

Conclusion

What is most noteworthy about LG is its versatility. We have used it to provide useful visualizations of classic rank data MDU, roll call data, multinomial regression and log-linear models and even social network data. Applications with many other types of data are not difficult to conceive.

This versatility springs from features of LG that are characteristic of De Leeuw's work: the generality of the indicator matrix method of data representation, a deep interest in finding unifying mathematical principles among varying techniques and an accompanying interest in parsimonious analysis. Contrast LG to WNOMINATE or the log-linear distance association methods or even to latent space social network modeling and we see that its relative simplicity is quite striking. Yet, it appears to give at least equally valuable and, in some cases, equivalent visualization of datasets. This is because, though it is simpler, LG expresses what is at the mathematical heart of these methods.

It can be countered that these other methods are superior since they can be used for inference and can account for model-fitting. Our response is that LG is an exploratory method and should be evaluated as such. Though LG does involve a probabilistic metric and an MLE approach to optimization, it should be recalled that these are expressly used as principles of construction, not of data generation. So, when we use LG to visualize multinomial regression or log-linear modeling results, for example, we do not mean to replace these by LG, but rather that they be used in tandem. Their predictive capabilities can be availed of as appropriate.

To paraphrase what De Leeuw (1988) [13] once said about homogeneity analysis, with LG we are interested in making a picture of our data which can be used to assist researchers in understanding it and making appropriate generalizations and predictions from it. It is for the statistician to construct the picture using a sensible process and to explain the implications of that process. It is for the researcher to do the predicting and generalizing.

This is not a new debate in any sense, but one that has gone on through much of the recent history of multivariate analysis, particularly where social science data or, more generally, categorical data is involved. In researching this dissertation, we came across an interesting instance of it involving the model-based MDS of Ramsay (1978) [54]. Ramsay had provided an approach to finding confidence region estimations for the points in an MDS solution. The model posited that observed distances between objects are distributed log-normally about the true distance with a variance dependent on point location. MLE methods were used to find point coordinates and it was shown that a generalized inverse of the information matrix gave the variances and covariances needed to find the elliptical confidence regions. It is not an overstatement to say that this is quite an impressive paper.

Ramsay (1982) [55] was invited to present this work to the Royal Statistical Society. The member responses to his paper are nearly as interesting as the paper itself and we briefly discuss some of them here to place our comments on LG into some historical context. A number of members believed that Ramsay's approach to MDS added little of value to the method. His distributional assumption seemed to be made for mathematical convenience. Even given this, the mathematics seemed needlessly complicated. It was doubted that social scientists, who at the time were the main users of MDS, could make much of it beyond what they could already do with classical MDS. If a useful MDS plot was produced but confidence regions were not well-behaved, what would that signify? Should the MDS result be seriously called into question?

Interestingly, De Leeuw is one of these respondents. While clearly respecting Ramsay's intellectual accomplishment, he weighs in on the side of the divide holding that MDS is best seen as a non-inferential, graphical method and is none the less valuable for it. So, we take a similar view with regard to LG. We do not dismiss the interesting model-based work we have reviewed, but do not view it as a slight to consider LG as an exploratory graphical method.

Another aspect of Ramsay's work which was criticized at the RSS was his use of MLE methodology in connection with MDS. Our study of LG gives some insight into this issue. Notice that, in LG, the likelihood of a configuration is a function of its scale. Given a suitable configuration, we can use dilation to increase its likelihood. In fact, for separated data, the likelihood can be made arbitrarily close to 1. Every configuration dilated in this way is exactly congruent to the starting configuration. They give precisely the same view of the data. And this is true no matter how the configuration is constructed. It could, for example, be the result of HOMALS. The likelihood interpretation arises simply by placing the LG metric on the plot. So, we have configurations quite possibly with low likelihood giving useful insight into the relationships between the objects, variables, and categories of our dataset. This is somewhat of a difficult concept for traditional views of MLE. Does likelihood in distance association methods come down to simply a choice of transformations of distances? In LG, all of this is fairly easily reconciled. MLE methodology is used only as a principle of construction or classification. It is not intended to give insight into the data generation processes or to give any deeper meaning to the space in which the LG plot is displayed.

It is the use of MLE abstracted from its usual interpretations that leads to the versatility of LG. Through this innovative blending of MLE, majorization, MDU, and general distance association methods, we have in LG a somewhat rare creation: a very useful technique for analysis of a wide range of data types that, at the same time, raises interesting questions at the foundations of its methods.

References

[1] A Agresti. *Categorical Data Analysis.* New York: John Wiley & Sons, second edition, 2002.

[2] A Agresti. *Analysis of Ordinal Categorical Data.* Hoboken: John Wiley & Sons, second edition, 2010.

[3] A Albert and JA Anderson. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71(1):1–10, 1984.

[4] F Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACS Computing Surveys*, 23(3):345–405, 1991.

[5] D Bohning and BG Lindsay. Monotonicity of quadratic-approximation algorithms. *Ann. Inst. Statist. Math.*, 40(4):641–663, 1988.

[6] I Borg and PJF Groenen. *Multidimensional Scaling: Theory and Applications.* New York: Springer-Verlag, second edition, 2005.

[7] I Borg and D Leutner. Measuring the similarity of MDS configurations. *Multivariate Behavioral Research*, 20(3):325–334, 1985.

[8] F Cailliez. The analytical solution of the additive constant problem. *Psychometrica*, 48(2):305–308, 1983.

[9] C Coombs. *Theory of Data.* New York: John Wiley & Sons, 1964.

[10] L Cooper. A new solution to the additive constant problem in metric multidimensional scaling. *Psychometrika*, 37(3):311–322, 1972.

[11] RJ de Ayala. *The Theory and Practice of Item Response Theory.* New York: Guilford Press, 2009.

[12] J de Leeuw. Convergence of the majorization method for multidimensional scaling. *Journal of Classification*, 5:163–180, 1988.

[13] J de Leeuw. Models and techniques. *Statistica Neerlandica*, 42:91–98, 1988.

[14] J de Leeuw. Principal component analysis of binary data by iterated singular value decomposition. *UCLA Department of Statistics Preprints*, 2004.

[15] J de Leeuw. Gifi goes logistic. *SCASA Keynote, UCLA Department of Statistics Preprints*, 2005.

[16] J de Leeuw. Logistic unfolding. *UCLA Department of Statistics Preprints*, 2005b.

[17] J de Leeuw. Majorization algorithms for distance association models. *UCLA Department of Statistics Preprints*, 2006.

[18] J de Leeuw. Principal component analysis of binary data by iterated singular value decomposition. *Computational Statistics & Data Analysis*, 50(1):21–39, 2006.

[19] J de Leeuw. On degenerate nonmetric unfolding solutions. *UCLA Department of Statistics Preprints*, 2007.

[20] J de Leeuw. Accelerating majorization algorithms. *UCLA Department of Statistics Preprints*, 2008.

[21] J de Leeuw. Block relaxation algorithms in statistics. *Information Systems and Data Analysis*, pages 308–331, Springer, 1994.

[22] J de Leeuw and W Heiser. Theory of multidimensional scaling. *Handbook of Statistics*, 2:285–316, 1982.

[23] J de Leeuw and K Lange. Sharp quadratic majorization in one dimension. *Computational Statistics and Data Analysis*, 53:2471–2479, 2009.

[24] J de Leeuw and P Mair. Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software*, 31(3), 2009.

[25] M de Rooij. Distance models for transition frequency data. *Ph.D Dissertation (Leiden University)*, 2001.

[26] M de Rooij and WJ Heiser. Graphical representations and odds ratios in a distance-association model for the analysis of cross-classified data. *Psychometrica*, 70(1):99–122, 2005.

[27] P Diaconis. Theories of data analysis: From magical thinking through classical statistics. *Exploring Data Tables, Trends and Shapes*, pages 1–37, 1985.

[28] JJ Faraway. *Extending the Linear Model with R.* Boca Raton, FL: Chapman & Hall, 2006.

[29] P Fox. *The PORT Mathematical Subroutine Library, Version 3.* Murray Hill, NJ: AT&T Bell Laboratories URL http://www.bell-labs.com/project/PORT/, 1997.

[30] T Fruchterman and E Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(1):1129–1164, 1991.

[31] W Gay. Usage summaries for selected optimization routines. *ATT Bell Laboratories Computing Science Technical Report (URL http://netlib.bell-labs.com/cm/cs/cstr/153.pdf)*, 1(153), 1990.

[32] J Gentle. *Matrix Algebra.* New York: Springer-Verlag, 2007.

[33] A Gifi. *Nonlinear Multivariate Analysis.* New York: John Wiley & Sons, 1990.

[34] LA Goodman. Simple models for the analysis of association in cross-classifications having ordered categories. *Journal of the American Statistical Association*, 74(367):537–552, 1979.

[35] LA Goodman. Association models and canonical correlation in the analysis of cross-classifications having ordered categories. *Journal of the American Statistical Association*, 76(374):320–334, 1981.

[36] LA Goodman. The analysis of dependence in cross-classifications having ordered categories, using log-linear models for frequencies and log-linear models for odds. *Biometrics*, 39(1):149–169, 1983.

[37] P Green and V Rao. *Applied Multidimensional Scaling.* New York: Holt, Rinehart and Winston, 1972.

[38] JP Guilford. *Psychometric Methods.* Cambridge, MA: Harvard University Press, 1936.

[39] J Hagenaars. *Categorical Longitudinal Data : log-linear panel, trend, and cohort analysis.* Newbury Park, CA: Sage Publications, 1990.

[40] WJ Heiser and J Meulman. Analysis of rectangular tables by joint and constrained multidimensional scaling. *Journal of Econometrics*, 22:139–167, 1986.

[41] PD Hoff and AE Raftery MS Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[42] C Kadushin. Friendship among the French financial elite. *American Sociological Review*, 60(2):202–221, 1995.

[43] T Kamada and S Kawai. An algorithm for drawing general undirected graphs. *Information Processing*, 31(1):7–15, 1989.

[44] M Kaufmann and D Wagner (eds.). *Drawing Graphs: Methods and Models.* New York: Springer-Verlag, 2001.

[45] K Lange. *Optimization.* New York: Springer-Verlag, 2004.

[46] S Loisel and Y Takane. Minimum polynomial extrapolation in MATLAB and in R. *Heriot-Watt University Department of Mathematics Preprints*, 2011.

[47] AE Maxwell. Canonical variate analysis when the variables are dichotomous. *Educational and Psychological Measurement*, 21:259–271, 1961.

[48] S Messick and R Abelson. The additive constant problem in multidimensional scaling. *Psychometrika*, 21(1):1–15, 1956.

[49] G Michailidis and J de Leeuw. The GIFI system of descriptive multivariate analysis. *Statistical Science*, 13(4):307–336, 1998.

[50] BL Nelson. *Stochastic Modeling: Analysis and Simulation.* Mineola, NY: Dover, 1995.

[51] J Padgett and C Ansell. Robust action and the rise of the Medici, 1400-1434. *American Journal of Sociology*, 98(6):1259–1319, 1993.

[52] K Poole and J Lewis J Lo R Carroll. Scaling roll call votes with `wnominate` in R. *Journal of Statistical Software*, 40(14):1–37, 2011.

[53] K Poole and H Rosenthal. A spatial model for legislative roll call analysis. *American Journal of Political Science*, 29(2):357–384, 1985.

[54] J O Ramsay. Confidence regions for multidimensional scaling analysis. *Psychometrika*, 43(2):145–160, 1978.

[55] J O Ramsay. Some statistical approaches to multidimensional scaling data. *Journal of the Royal Statistical Society*, 145(3):285–312, 1982.

[56] G Rosman and A Bronstein M Bronstein A Sidi R Kimmel. Fast multidimensional scaling using vector extrapolation. *SIAM Journal of Scientific Computing*, 2, 2008.

[57] PH Schonemann. On metric multidimensional unfolding. *Psychometrica*, 35(3):349–366, 1970.

[58] RN Shepard. Stimulus and response generalizations: A stochastic model relating generalization to distance in psychological space. *Psychometrica*, 22(4):325–345, 1957.

[59] D Smith and W Ford A Sidi. Extrapolation methods for vector sequences. *SIAM Review*, 29(2):199–233, 1987.

[60] I Spence. A MONTE CARLO evaluation of three nonmetric multidimensional scaling algorithms. *Psychometrica*, 37(4):461–486, 1972.

[61] Y Takane. Analysis of contingency tables by ideal point discriminant analysis. *Psychometrica*, 52(4):493–513, 1989.

[62] Y Takane. Ideal point discriminant analysis and ordered response categories. *Behaviormetrica*, 1(26):31–46, 1989.

[63] Y Takane and H Bozdogan T Shibayama. Ideal point discriminant analysis. *Psychometrica*, 52(3):371–392, 1987.

[64] WS Torgerson. Multidimensional scaling: I. theory and method. *Psychometrica*, 48(3):401–419, 1952.

[65] GJ Upton. *The Analysis of Cross-tabulated Data*. New York: John Wiley & Sons, 1978.

[66] WN Venables and BD Ripley. *Modern Applied Statistics with S-Plus*. New York: Springer-Verlag, 2002.

[67] S Wasserman and K Faust. *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press, 1994.