Feature Network Models for Proximity Data

Frank, Laurence Emmanuelle,
Feature Network Models for Proximity Data. Statistical inference, model selection, network representations and links with related models.
Dissertation Leiden University - With ref. - With summary in Dutch.
Subject headings: additive tree; city-block models; distinctive features models; feature models; feature network models; feature selection; Monte Carlo simulation; statistical inference under inequality constraints.

ISBN 90-8559-179-1

Printed by Optima, Rotterdam

Manuscript prepared in LaTeX (pdftex) with the TeX previewer TeXShop (v1.40), using the memoir document class (developed by P. Wilson) and the apacite package for APA style bibliography (developed by E. Meijer).

Feature Network Models for Proximity Data

Statistical inference, model selection, network representations and links with related models

PROEFSCHRIFT

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van de Rector Magnificus Dr. D.D. Breimer,
hoogleraar in de faculteit der Wiskunde en
Natuurwetenschappen en die der Geneeskunde,
volgens besluit van het College voor Promoties
te verdedigen op donderdag 21 september 2006
klokke 13.45 uur

door

Laurence Emmanuelle Frank

geboren te Delft
in 1969

PROMOTIECOMMISSIE

Promotor:	Prof. Dr. W. J. Heiser
Referent:	Prof. J. E. Corter, Ph.D., Columbia University, New York, USA
Overige leden:	Prof. Dr. I. Van Mechelen, K.U. Leuven, België Prof. Dr. V. J. J. P. van Heuven Prof. Dr. J. J. Meulman Prof. Dr. P. M. Kroonenberg

To my parents

"On ne peut se flatter d'avoir le dernier mot d'une théorie, tant qu'on ne peut pas l'expliquer en peu de paroles à un passant dans la rue."

[It is not possible to feel satisfied at having said the last word about some theory as long as it cannot be explained in a few words to any passer-by encountered in the street.]

Joseph Diaz Gergonne, French mathematician (Chasles, 1875, p. 115).

Acknowledgements

A large number of persons contributed in several ways to this dissertation and I am indebted to them for their support.

I learned a lot about research in psychometrics from being a member of the Interuniversity Graduate School for Psychometrics and Sociometrics (IOPS). I would like to thank the IOPS-students for the agreeable time and a special thank to Marieke Timmerman, for her interest and the pleasant conversations. A special thank also to Susañña Verdel for the enjoyable way we prepared the IOPS meetings, and for the wonderful way she organizes all practical IOPS-issues, always trying to offer the best possible conditions for staff and students.

I am very grateful for the opportunities given to me to attend conferences of the Psychometric Society and the International Federation of Classification Societies, which introduced me to the scientific community of our field of research and has been very inspiring for my own research.

I am greatly indebted to Prof. L.J. Hubert, Ph.D. (University of Illinois at Urbana-Champaign, USA) for helping me to implement the Dykstra algorithm in Matlab during his stay in Leiden, and for his useful comments on earlier versions of the second and third chapter of this dissertation.

For support on a more daily basis I would like to thank my colleagues of Psychometrics and Research Methodology and Data Theory Group (Universiteit Leiden): Bart Jan van Os for helping me with technical issues at crucial points during this research project, but also for his interest and the pleasant coffee breaks; Mark de Rooij for showing me how to do research in our field and how to accomplish a Ph.D. project; Mariëlle Linting for sharing a lot of nice conference experiences and hotel rooms during the whole project; Matthijs Warrens for the inspiring conversations about research; Marike Polak for the daily pleasant, encouraging conversations with lots of coffee and tea, and her sincere interest, which also holds for Rien van der Leeden.

The accomplishment of this dissertation would not have been possible without the love and support of my family and friends. To all who supported me during these years: many thanks for your friendship and all the joyous moments shared. It helped me to place this work in the right perspective.

Contents

List of Figures

List of Tables

Notation and Symbols

Notation conventions

matrices:	bold capital
vectors:	bold lowercase
scalars, integers:	lowercase

Symbols

Symbol	*Description*
O	an object or stimulus
m	the number of objects, stimuli
i	index $i = 1, \cdots, m$
j	index $j = 1, \cdots, m$
k	index $k = 1, \cdots, m$
n	the number of object pairs $= \frac{1}{2}m(m-1)$
l	index $l = 1, \cdots, n$
N	the number of replications of samples of size $n \times 1$
ℓ	index $\ell = 1, \cdots, N$
f	a frequency value associated with an object pair
δ	a dissimilarity value associated with an object pair
$\hat{\delta}$	an estimated dissimilarity value associated with an object pair
$\boldsymbol{\delta}$	an $n \times 1$ vector with dissimilarities between all object pairs
$\hat{\boldsymbol{\delta}}$	an $n \times 1$ vector with estimated dissimilarities between all object pairs
$\boldsymbol{\Delta}$	an $m \times m$ matrix with dissimilarities
$\widetilde{\Delta}_{l\ell}$	a random variable producing realisations $\tilde{\delta}_{l\ell}$
$\tilde{\delta}_{l\ell}$	a realisation of random variable $\widetilde{\Delta}_{l\ell}$
$\widetilde{\boldsymbol{\Delta}}$	an $n \times N$ matrix of random variables $\widetilde{\Delta}_{l\ell}$
$\overline{\Delta}_l$	mean of a row l of $\widetilde{\boldsymbol{\Delta}}$
ς	a similarity value associated with an object pair
$\boldsymbol{\varsigma}$	an $n \times 1$ vector with similarities between all object pairs
$\boldsymbol{\Sigma}_{\varsigma}$	an $m \times m$ matrix with similarities
F	a feature, which is a binary $(0, 1)$ vector of size $m \times 1$
F_{C}	a *cluster* feature, which is a binary $(0, 1)$ vector of size $m \times 1$
F_{U}	a *unique* feature, which is a binary $(0, 1)$ vector of size $m \times 1$
T	the number of features

T_{C}	the number of *cluster* features
T_{U}	the number of *unique* features
$T_{\mathcal{D}}$	the total number of distinctive features $= \frac{1}{2}(2^m) - 1$
t	index for the features: $t = 1, \cdots, T$
t_{C}	index for the *cluster* features: $t_{\mathrm{C}} = 1, \cdots, T_{\mathrm{C}}$
t_{U}	index for the *unique* features: $t_{\mathrm{U}} = 1, \cdots, T_{\mathrm{U}}$
$\mathcal{S}_i$	the set of features that represents object O_i
$\mathbf{E}$	an $m \times T$ matrix with columns representing features
$\mathbf{e}$	a row vector from the matrix $\mathbf{E}$
e	an element of the matrix $\mathbf{E}$
$\mathbf{E}_T$	an $\mathbf{E}$ matrix with special feature structure that yields a tree representation
$\mathbf{E}_C$	the part of $\mathbf{E}_T$ (size $m \times T_{\mathrm{C}}$) that represents the set of *cluster* features
$\mathbf{E}_U$	the part of $\mathbf{E}_T$ (size $m \times T_{\mathrm{U}}$) that represents the set of *unique* features
$\mathbf{X}$	an $n \times T$ matrix with featurewise distances obtained with $\mathbf{x}' = \lvert\mathbf{e}_{it} - \mathbf{e}_{jt}\rvert$
$\mathbf{x}'$	a row vector from the matrix $\mathbf{X}$
$\mathbf{x}$	a column vector from the matrix $\mathbf{X}$
$\mathbf{X}_T$	an $n \times T_{\mathrm{C}} + T_{\mathrm{U}}$ matrix with featurewise distances obtained with $\mathbf{E}_T$
$\mathcal{D}$	the complete set of featurewise distances
d	a distance between an object pair
$\mathbf{d}$	an $n \times 1$ vector of distances between all object pairs
$\hat{\mathbf{d}}$	an $n \times 1$ vector of estimated distances between all object pairs
$\hat{\mathbf{d}}_{\mathrm{T}}$	an $n \times 1$ vector of estimated distances between all object pairs for a tree structure
η	feature discriminability parameter
η_{OLS}	true value of ordinary least squares feature discriminability parameter
η_{ICLS}	true value of inequality constrained least squares feature discriminability parameter
η_{L}	true value of Lasso feature discriminability parameter
η_{PL}	true value of Positive Lasso feature discriminability parameter
$\boldsymbol{\eta}$	an $T \times 1$ vector of feature discriminability parameters
$\boldsymbol{\eta}_{\mathrm{OLS}}$	an $T \times 1$ vector of true values η_{OLS}
$\boldsymbol{\eta}_{\mathrm{ICLS}}$	an $T \times 1$ vector of true values η_{ICLS}
$\boldsymbol{\eta}_{\mathrm{L}}$	an $T \times 1$ vector of true values η_{L}
$\boldsymbol{\eta}_{\mathrm{PL}}$	an $T \times 1$ vector of true values η_{PL}
$\hat{\eta}, \hat{\eta}_{\mathrm{OLS}}$	estimated values of η, η_{OLS}, η_{ICLS}, η_{L}, η_{PL}
C	the number of constraints necessary to obtain $\hat{\boldsymbol{\eta}}_{\mathrm{ICLS}}$
c	index $c = 1, \cdots, C$
$\mathbf{r}$	a $C \times 1$ vector with constraints
$\mathbf{A}$	a $C \times T$ matrix of constraints of rank c
$\boldsymbol{\lambda}_{\mathrm{KT}}$	a $m \times 1$ vector with Kuhn-Tucker mutipliers
$\boldsymbol{\epsilon}$	a $n \times 1$ vector with error values ($\boldsymbol{\epsilon} = \boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta}$)
$\hat{\boldsymbol{\epsilon}}$	a $n \times 1$ vector with estimated error values ($\hat{\boldsymbol{\epsilon}} = \boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}$)
$\hat{\epsilon}$	an element from the vector $\hat{\boldsymbol{\epsilon}}$
σ^2, σ	true variance and standard deviation of $\boldsymbol{\epsilon}$
$\hat{\sigma}^2, \hat{\sigma}$	estimated variance and standard deviation of $\hat{\boldsymbol{\epsilon}}$
$\sigma^2_\eta, \sigma_\eta$	true variance and standard error of η
$\hat{\sigma}^2_\eta, \hat{\sigma}_\eta$	estimated nominal variance and estimated nominal standard error of η
$\hat{\sigma}^2_{\hat{\eta}}, \hat{\sigma}_{\hat{\eta}}$	estimated nominal variance and nominal standard error of $\hat{\eta}$
$\sigma^2_{\mathrm{OLS}}, \sigma_{\mathrm{OLS}}$	true variance and standard error of $\hat{\eta}_{\mathrm{OLS}}$

$\hat{\sigma}^2_{\text{OLS}}$, $\hat{\sigma}_{\text{OLS}}$	estimated variance and standard error of $\hat{\eta}_{\text{OLS}}$
σ^2_{ICLS}, σ_{ICLS}	true variance and standard error of $\hat{\eta}_{\text{ICLS}}$
$\hat{\sigma}^2_{\text{ICLS}}$, $\hat{\sigma}_{\text{ICLS}}$	estimated variance and standard error of $\hat{\eta}_{\text{ICLS}}$
B	number of bootstrap samples
b	index $b = 1, \cdots, B$
$\mathbf{b}_b$	a bootstrap sample ($n \times 1$ vector)
$\mathbf{b}^*_b$	a bootstrap sample, multivariate
$\tilde{\mathbf{b}}_b$	a bootstrap sample, with sampled residuals
sd_B	standard deviation of B bootstrap samples
S	number of simulation samples
a	index $a = 1, \cdots, S$
$\mathbf{s}^*$	a simulation sample ($n \times 1$ vector)
κ, p	parameters binomial distribution
GCV	generalized cross-validation statistic
GCV_{FNM}	GCV using inequality constrained least squares estimation

Chapter 1

Introducing Feature Network Models

Feature Network Models (FNM) are graphical models that represent dissimilarity data in a discrete space with the use of features. Features are used to construct a distance measure that approximates observed dissimilarity values as closely as possible. Figure 1.1 shows a feature network representation of all presidents of the United States of America, based on 14 features, which are binary variables that indicate

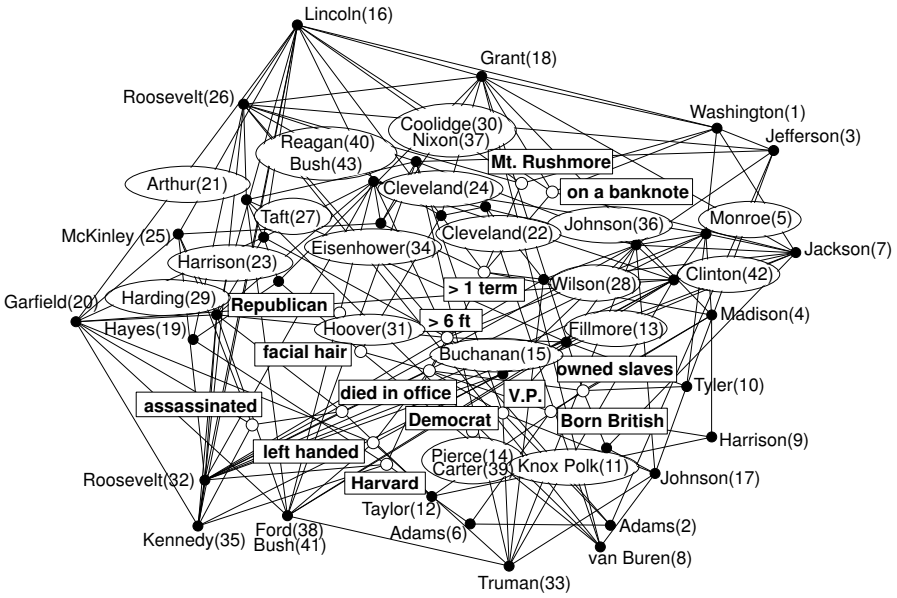

Figure 1.1: Feature network of all presidents of the USA based on 14 features from Schott (2003, pp. 14-15). The presidents are represented as vertices (black dots) and labeled with their names and chronological number. The features are represented as internal nodes (white dots).

whether a president has the characteristic or not (the characteristics were adapted from Schott, 2003, pp. 14-15). The features are: political party, whether the president served more than 1 term, was assassinated or died in office, was taller than 6 ft, served as vice-president (V.P.), had facial hair, owned slaves, was born British, appeared on a banknote, is represented at Mount Rushmore, went to Harvard and is left handed.

In the network the presidents are represented as vertices (black dots) labeled with their name and chronological number. The features are represented as internal nodes (white dots). The general idea of a network representation is that an edge between objects gives an indication of the relation between the objects. For example, there is an edge present between president Kennedy and the feature *assassinated*, which in turn has a direct link with the feature *died in office*. As a result of the embedding of this 14-dimensional structure in 2-dimensional space, objects that are close to each other in terms of distances are more related to each other than objects that are further apart. The network representation has a rather complex structure with a large number of edges and is not easily interpretable. One of the objectives of Feature Network Models is to obtain a parsimonious network graph that adequately represents the data. The three components of FNM, the features, the network representation and the model, will be explained successively in this introduction, using a small data set with16 objects and 8 features, that can be adequately represented in 2 dimensions. While explaining the different components, the topics of the chapters of this monograph will be introduced.

1.1 Features

A feature is, in a dictionary sense, a prominent characteristic of a person or an object. In the context of FNM, a feature is a binary (0,1) vector that indicates for each object or stimulus in an experimental design whether a particular characteristic is present or absent. Features are not restricted to nominal variables, like eye color, or binary variables as voiced versus unvoiced consonants. Ordinal and interval variables, if categorized, can be transformed into a set of binary vectors (features) using dummy coding. Table 1.1 shows an example of features deriving from an experimental design created by Tversky and Gati (1982). The stimuli are 16 types of plants that vary depending on the combination of two qualitative variables, the form of the ceramic pot (4 types) and the elongation of the leaves of the plants (4 types), see Figure 1.2. The two variables can be represented as features using dummy coding for the levels of each variable and Table 1.1 shows the resulting feature matrix. In the original experiment, all possible pairs of stimuli were presented to 29 subjects who were asked to rate the dissimilarity between each pair of stimuli on a 20-point scale. The data used for the analyses are the average dissimilarity values over the 29 subjects as presented in Gati and Tversky (1982, Table 1, p. 333).

In psychology, the concept of feature as basis for a model has been introduced by Tversky (1977) who proposed the Contrast Model, which is a set-theoretical approach where objects are characterized by subsets of discrete features and similarity between objects is described as a comparison of features. The Contrast Model

Table 1.1: Feature matrix of 16 plants (Figure 1.2) varying in form of the pot (features: a, b, c) and elongation of the leaves (features: p, q, r), see Tversky and Gati (1982).

Plants		Features					
		a	b	c	p	q	r
1	ap	1	0	0	1	0	0
2	aq	1	0	0	0	1	0
3	ar	1	0	0	0	0	1
4	as	1	0	0	0	0	0
5	bp	0	1	0	1	0	0
6	bq	0	1	0	0	1	0
7	br	0	1	0	0	0	1
8	bs	0	1	0	0	0	0
9	cp	0	0	1	1	0	0
10	cq	0	0	1	0	1	0
11	cr	0	0	1	0	0	1
12	cs	0	0	1	0	0	0
13	dp	0	0	0	1	0	0
14	dq	0	0	0	0	1	0
15	dr	0	0	0	0	0	1
16	ds	0	0	0	0	0	0

was intended as an alternative to the dimensional and metric methods like multidimensional scaling, because Tversky questioned the assumptions that objects can be adequately represented as points in some coordinate space and that dissimilarity behaves like a metric distance function. Believing that it is more appropriate to represent stimuli in terms of many qualitative features than in terms of a few quantitative dimensions, Tverksy proposed a set-theoretical approach where objects are characterized by subsets of discrete features and similarity between objects is described as a comparison of features. According to Tversky, the representation of an object as a collection of features parallels the mental process of participants faced with a comparison task: participants extract and compile from their data base of features a limited list of relevant features on the basis of which they perform the required task by feature matching. This might lead to a psychologically more meaningful model since it is testing some possible underlying processes of similarity judgments.

Distinctive features versus common features

The Contrast Model describes the similarity between two objects in terms of a linear combination of the features they share (the *common features*) and the features that distinguish between them (*distinctive features*). The idea is that the similarity between two objects increases with addition of common features and/or deletion of distinctive features. In set-theoretical terms, a *common feature* is equal to the intersection of the feature sets that belong to each pair of objects and a *distinctive feature* is equal to the union minus the intersection of the feature sets (= the symmetric set difference).

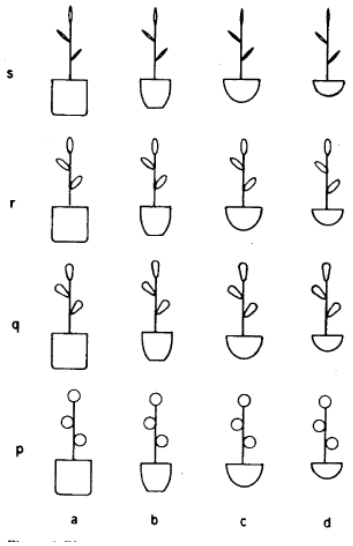

Figure 1.2: Experimental conditions *plants* data. The 16 plants vary in the form of the pot and in elongation of the leaves. (Adapted with permission from: Tversky and Gati (1982), Similarity, separability, and the triangle inequality. *Psychological Review, 89*, 123-154, published by APA.)

For example plant 1 (Table 1.1) is characterized by the feature set $\mathcal{S}_1 = \{a, p\}$ and plant 2 is characterized by the feature set $\mathcal{S}_2 = \{a, q\}$. The plants 1 en 2 have one common feature $\{a\}$ and two distinctive features $\{p, q\}$. The mathematical representation of the similarity between the plants 1 and 2 following the Contrast Model is equal to:

$$\varsigma(\mathcal{S}_1, \mathcal{S}_2) = \theta f(\mathcal{S}_1 \cap \mathcal{S}_2) - \alpha f(\mathcal{S}_1 - \mathcal{S}_2) - \beta f(\mathcal{S}_2 - \mathcal{S}_1), \tag{1.1}$$

where the first part after the equal sign represents a function f of the common features part with the corresponding weight θ and the remaining two parts express functions of the distinctive features part with corresponding weights α and β, and the total similarity value is expressed as a linear combination of the common features part and the distinctive features part. Tversky (1977) and Gati and Tversky (1984) observed that the relative weight of distinctive features and common features varies with the nature of the task: in conceptual comparisons, the relative weight of common to distinctive features was higher in judgments of similarity than in judgments of dissimilarity. The relative weight of common to distinctive features also changed depending on the task instructions: when subjects were instructed to rate the amount in which to objects differ, the relative weight of the distinctive features to common features increases.

Table 1.2: Overview of graphical and non-graphical models based on common features (CF) and distinctive features (DF)

Model	Author(s)	CF, DF	Graphical representation
Contrast Model (CM)	Tversky (1977)	CF + DF	not available
Additive similarity trees	Sattath and Tversky (1977)	DF	additive tree
ADCLUS	Shepard and Arabie (1979)	CF	clusters with contour lines
MAPCLUS	Arabie and Carroll (1980)	CF	clusters with contour lines
EXTREE	Corter and Tversky (1986)	DF	additive tree + marked segments
CLUSTREES	Carroll and Corter (1995)	CF	trees like MAPCLUS and EXTREE
Feature Network Models (FNM)	Heiser (1998)	DF	network (trees)
Modified Contrast Model (MCM)	Navarro and Lee (2004)	CF + DF	clusters

The Contrast Model in its most general form has been used in practice with a priori features only (Gati & Tversky, 1984; Keren & Baggen, 1981; Takane & Sergent, 1983), but many models have been developed since, which search for either the common features part or the distinctive features part of the model, or a combination of both. The models that are based uniquely on common features, the *common features models* are several versions of additive clustering: ADCLUS (Shepard & Arabie, 1979), MAPCLUS (Arabie & Carroll, 1980) and CLUSTREES (Carroll & Corter, 1995). It should be noted that the CLUSTREES model differs from the other common features models because it finds distinctive feature representations of common features models. The additive similarity trees (Sattath & Tversky, 1977)) and the extended similarity trees (EXTREE, Corter & Tversky, 1986) both use distinctive features and are *distinctive features models*. A model that has the closest relation to the Contrast Model is the Modified Contrast Model developed by Navarro and Lee (2004) that aims at finding a set of both common and distinctive a priori unknown features that best describes the data. Table 1.2 gives an overview of the models with the corresponding graphical representation, which will be explained in Section 1.2.

Feature Network Models (FNM) use the set-theoretical approach proposed by Tversky, but are restricted to distinctive features. The definition of distinctive features used in FNM states that features are not inherently distinctive, but become distinctive after application of the set-theoretic transformation, in this case, the symmetric set difference. This definition means that it is not possible to classify, for

example, the features describing the plants in Table 1.1 as distinctive or common because the set-theoretic transformations have not taken place yet. Chapter 4 makes the definition of distinctive feature more concrete by defining the complete set of distinctive features and by showing how to generate the complete set in an efficient way using a special binary code, the Gray code.

Although the two types of feature models, the common features model (CF) and the distinctive features model (DF), are in a sense opposed to each other and can function as separate models, there is a clear relation between the two. Sattath and Tversky (1987), and later Carroll and Corter (1995), have demonstrated that the CF model can be translated into the DF model and vice versa. However, these theoretical results have not been applied in the practice of data analysis, where one fits either one of the two models, or the combination of both. Chapter 5 adds an important result to the theoretical translation between the CF model and the DF model, and shows the consequences for the practice of data analysis. It will become clear that for any fitted CF model it is possible to find an equally well fitting DF model with the same shared features (common features) and feature weights, and with the same number of independent parameters. Following the same results, a model that combines the CF and DF models can be expressed as a combination of two separate DF models.

Where do features come from?

The features in Table 1.1 are a direct result of the experimental design and represent the physical characteristics of the objects (the plants). Most of the feature methods mentioned in the previous sections use a priori features that derive from the experimental design or a psychological theory. In the literature, examples where features are estimated from the data are rare. There is, however, no necessary relation between the physical characteristics that are used to specify the objects and the psychological attributes that subjects might use when they perceive the objects. It is therefore useful to estimate the features from the data as well. An example of a data analysis with theoretic features and with features estimated from the data will be given for the *plants* data. Chapter 4 is entirely devoted to the subject of selecting adequate subsets of features resulting from theory or estimated from the data.

It should be noted that a well known set of theoretic features plays an important role in phonetics as part of the Distinctive Feature Theory. The distinctive features form a binary system to uniquely classify the sounds of a language, the phonemes. The term distinctive used here is not the set-theoretic term used for the distinctive features of the FNM. Various sets of distinctive features have been proposed in phonetics and the first set consisting of 14 features has been proposed by Jakobson, Fant, and Halle (1965): the distinctive features are the ultimate distinctive entities of language since none of them can be broken down into smaller linguistic units (p. 3). A subset of these distinctive features will be used to illustrate the Feature Network Models in Chapter 2 and Chapter 4.

Feature distance and feature discriminability

FNM aim at estimating distance measures that approximate observed dissimilarity values as closely as possible. The symmetric set difference can be used as a distance measure between each pair of objects O_i and O_j that are characterized by the corresponding feature sets $\mathcal{S}_i$ and $\mathcal{S}_j$. Following Goodman (1951, 1977) and Restle (1959, 1961), a distance measure that satisfies the metric axioms can be expressed as a simple count μ of the elements of the symmetric set difference, a count of the non common elements between each pair of objects O_i and O_j and becomes the *feature distance*:

$$d(O_i, O_j) = \mu[(\mathcal{S}_i - \mathcal{S}_j) + (\mathcal{S}_j - \mathcal{S}_i)] = \mu[(\mathcal{S}_i \cup \mathcal{S}_j) - (\mathcal{S}_i \cap \mathcal{S}_j)]. \tag{1.2}$$

Heiser (1998) demonstrated that the feature distance in terms of set operations can be re-expressed in terms of coordinates and as such, is equal to a city-block metric on a space with binary coordinates, a metric also known as the *Hamming distance*. If $\mathbf{E}$ is a binary matrix of order $m \times T$ that indicates which of the T features describe the m objects, as in Table 1.1, the re-expression of the feature distance in terms of coordinates is as follows:

$$\begin{aligned} d(O_i, O_j) &= \mu[(\mathcal{S}_i \cup \mathcal{S}_j) - (\mathcal{S}_i \cap \mathcal{S}_j)] \\ &= \sum_t |e_{it} - e_{jt}|, \end{aligned} \tag{1.3}$$

where $e_{it} = 1$ if feature t applies to object i, and $e_{it} = 0$ otherwise. In the example of the plants 1 and 2 the feature distance is equal to the sum of the distinctive features $\{p, q\}$, in this case 2. The properties of the feature distance and especially the relation between the feature distance and the city-block metric are discussed in Chapter 5.

For fitting purposes, it is useful to generalize the distance in Equation 1.3 to a weighted count, i.e., the weighted feature distance:

$$d(O_i, O_j) = \sum_t \eta_t |e_{it} - e_{jt}|, \tag{1.4}$$

where the weights η_t express the relative contribution of each feature. Each feature splits the objects into two classes, and η_t measures how far these classes are apart. For this reason, Heiser (1998) called the feature weight a *discriminability parameter*. The feature discriminability parameters are estimated by minimizing the following least squares loss function:

$$\min_{\boldsymbol{\eta}} = \|\mathbf{X}\boldsymbol{\eta} - \boldsymbol{\delta}\|^2, \tag{1.5}$$

where $\mathbf{X}$ is of size $n \times T$ and $\boldsymbol{\delta}$ is a $n \times 1$ vector of dissimilarities, with n equal to all possible pairs of m objects: $\frac{1}{2}m(m-1)$. The problem in Equation 1.5 is expressed in a more convenient multiple linear regression problem, where the matrix $\mathbf{X}$ is obtained by applying the following transformation on the rows of matrix $\mathbf{E}$ for each pair of objects, where the elements of $\mathbf{X}$ are defined by:

$$x_{lt} = |e_{it} - e_{jt}|, \tag{1.6}$$

where the index $l = 1, \cdots, n$ varies over all pairs (i, j). The result is the binary $(0, 1)$ matrix $\mathbf{X}$, where each row represents the distinctive features for each pair of objects, with 1 meaning that the feature is distinctive for a pair of objects. It is important to notice that features become truly distinctive features only after this transformation, while the features in the matrix $\mathbf{E}$ are not inherently common or distinctive. The weighted sum of these distinctive features is the feature distance for each pair of objects and is equal to $\mathbf{d} = \mathbf{X}\boldsymbol{\eta}$. The feature distances serve as starting point for the construction of the network, as will become clear in the next section.

1.2 Feature Network

In general, there are two types of graphical representations of proximity data: spatial models and network models. The spatial models - multidimensional scaling - represent each object as a point in a coordinate space (usually Euclidean space) in such a way that the metric distances between the points approximate the observed proximities between the objects as closely as possible. In network models, the objects are represented as vertices in a connected graph, so that the spatial distances along the edges between the vertices in the graph approximate the observed proximities among the objects. In MDS, the primary objective is to find optimal coordinate values that lead to distances that approximate the observed proximities between the objects, whereas in network models, the primary objective is to find the correct set of *relations* between the objects that describe the observed proximities.

Parsimonious feature graphs

The symmetric set difference, which is the basis of FNM, describes the relations between the object pairs in terms of distinctive features and permits a representation of the stimuli as vertices in a network using the feature distance. In the network, called a *feature graph*, the structural relations between the objects in terms of distinctive features is expressed by edges connecting adjacent objects and the way in which the objects are connected depends on the fitted feature distances. Distance in a network is the path travelled along the edges; the distance that best approximates the dissimilarity value between two objects is the shortest path between the two corresponding vertices in the network.

The feature distance has some special properties resulting from its set-theoretical basis that allows for a representation in terms of shortest paths, which also considerably reduces the number of edges in the network. A complete network, i.e. a network where all pairs of vertices (representing the m objects) are connected has $n = \frac{1}{2}m(m-1)$ edges. Figure 1.3 shows a complete network of the *plants* data where all pairs of plants are connected with an edge. Such a network is obviously not adequate in explaining the relations between the objects, due to lack of parsimony.

The feature distance parallels the path-length distance in a valued graph when one of the metric axioms, the triangle inequality, becomes an equality: $d_{ik} = d_{ij} + d_{jk}$ (Flament, 1963; Heiser, 1998). In a network graph, each time that the distance d_{ik} is exactly equal to the sum $d_{ij} + d_{jk}$ the edge between the objects i and k can be

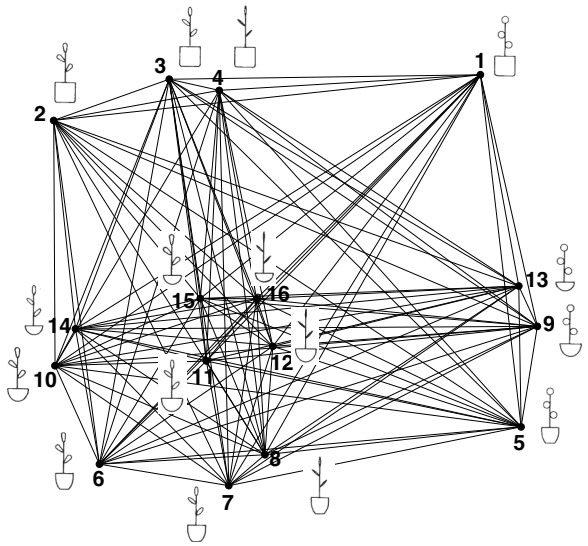

Figure 1.3: Complete network *plants* data.

excluded, resulting in a parsimonious subgraph of the complete graph. In terms of features the condition $d_{ik} = d_{ij} + d_{jk}$ is reached when object j is *between* objects i and k. The objects can be viewed as sets of features: $\mathcal{S}_i$, $\mathcal{S}_j$, and $\mathcal{S}_k$. Betweenness of $\mathcal{S}_j$ depends on the following conditions (Restle, 1959):

1. $\mathcal{S}_i$ and $\mathcal{S}_k$ have no common members which are not also in $\mathcal{S}_j$;
2. $\mathcal{S}_j$ has no unique members which are in neither $\mathcal{S}_i$ nor $\mathcal{S}_k$.

Apart from the experimental objects, we can also identify hypothetical objects called *internal nodes*. These are new feature sets defined in terms of the intersection of available feature sets. As an example, Figure 1.4 shows two of the plants (numbers 13 and 14, with feature sets $\{d, p\}$ and $\{d, q\}$, respectively) and an internal node defined by a feature set containing the single feature $\{d\}$. It is clear that betweenness holds with respect to the internal node, because its feature set is exactly equal to the intersection of the sets belonging to plants 13 and 14, as can be seen in the right part of Figure 1.4. For the network representation in terms of edges, the betweenness condition implies that the feature distances between the three objects reach the triangle equality condition. Calling the internal node *'dpot'*, we have $d_{14,13} = d_{14,dpot} + d_{dpot,13} = 1 + 1 = 2$. For the ease of explanation the feature distances are represented as unweighted counts of the number of distinctive features. Consequently, the edge between the plants 13 and 14 can be excluded (see the left part of Figure 1.4).

Hence, sorting out the additivities in the fitted feature distances, with the possible inclusion on internal nodes, and excluding edges that are sums of other edges

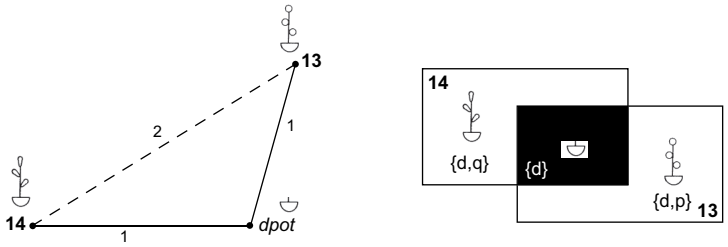

Figure 1.4: Triangle equality and betweenness.

results in a parsimonious subgraph of the complete graph, expressed in a binary adjacency matrix with ones indicating the presence of an edge. It should be noted that the approach of sorting out the additivities is different from the network models of Klauer (1989, 1994) and Klauer and Carroll (1989), who sort out the additivities on the observed dissimilarities. Using the fitted distances instead leads to better networks because the distances are model quantities whereas dissimilarities are subject to error. The network approach used in FNM is also different from the social network models (*cf.* Wasserman & Faust, 1994) that use the adjacency matrix as the starting point of the analyses, whereas for the FNM it is the endpoint.

Figure 1.5 shows the result of sorting out the triangle equalities on the fitted feature distances for the *plants* data, where features d and s have been omitted. Note that each of the first four features a, b, c, and d is redundant, since it is a linear combination of the other three. The same is true for p, q, r, and s. To avoid problems with multicollinearity in the estimation, one feature in each set has to be dropped. Hence from now on, we continue the example with a reduced set of 6 features. The *feature graph* clearly has gained in parsimony compared to the complete network in Figure 1.3. The network has been embedded in 2-dimensional Euclidean space, with the help of PROXSCAL (a multidimensional scaling program distributed as part of the Categories package by SPSS, Meulman & Heiser, 1999), allowing ratio proximity transformation. The edges in the network express the relations between pairs of plants based on the weighted sum of their distinctive features. More details on the interpretation of the network model will be given in section 1.3. Chapter 5 discusses in detail the betweenness condition as well as the algorithm used for sorting out the triangle equalities and also introduces the internal node as a way to simplify the network representation.

Embedding in low-dimensional space

A network is by definition coordinate-free because it is entirely determined by the presence or absence of edges between vertices and by the lengths of these edges. The distances between the objects in a network do not serve the same interpretational purpose as in multidimensional scaling, where distances are a direct expression of the strength of the relation between the objects. In FNM the relation between the

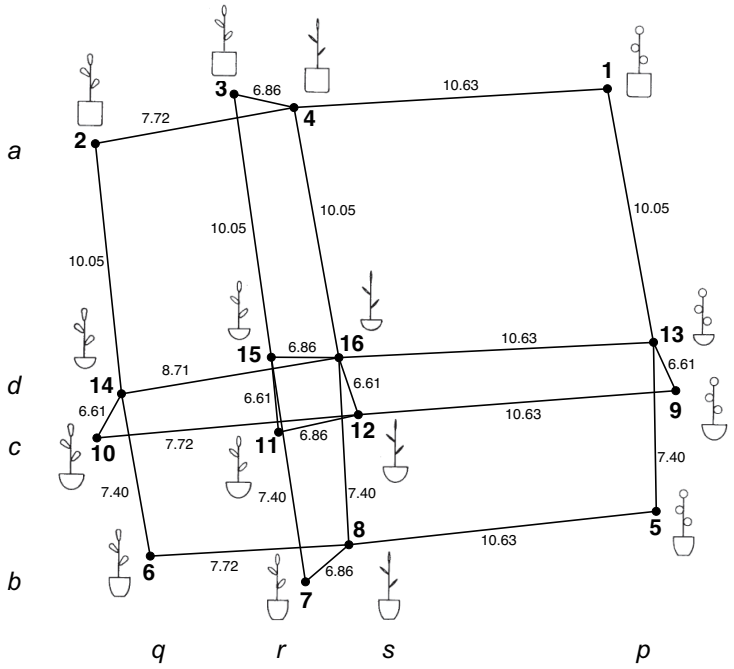

Figure 1.5: Feature graph of the *plants* data using the features resulting from the experimental design with varying elongation of leaves and form of the pot (with 6 of the 8 features).

objects is primarily expressed by the presence of edges between the vertices. The embedding of the network in a lower dimensional space is therefore of secondary importance. In this monograph, the embedding chosen for the feature graphs results from analysis with PROXSCAL (Meulman & Heiser, 1999) of the Euclidean distances computed on the weighted feature matrix. Most of the representations are in 2 dimensions, sometimes other options are chosen to obtain a representation that is better in terms of visual interpretability.

The embedding discussed so far concerns the vertices of the network representing the objects, while the features themselves can also be visualized. There are several possibilities to represent features graphically in the network plot. Most of the network plots in this monograph show the features as vectors. This representation is obtained using PROXSCAL (Meulman & Heiser, 1999) with the option of restricting the solution of the common space by a linear combination of the feature variables. Another version is to represent the features as internal nodes, as in the presidents network (Figure 1.1). The representation of features as internal nodes which will be discussed and illustrated in Chapter 5.

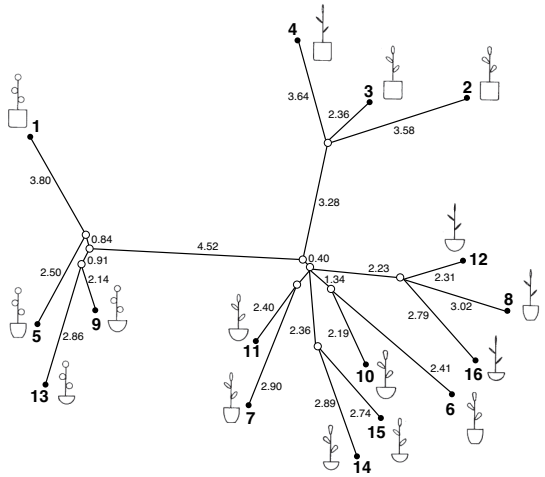

Figure 1.6: Additive tree representation of the *plants* data.

Feature structure and related graphical representation

The feature structure typically represented by FNM is a non-nested structure, or in terms of clusters, an overlapping cluster structure. In contrast, hierarchical trees and additive trees require a strictly nested feature structure. The graphical representation of a non-nested structure is more complex than a tree (Carroll & Corter, 1995). At least three solutions have been proposed in the literature (see the overview in Table 1.2): ADCLUS starts with a cluster representation and adds contour lines around the cluster to reveal the overlapping structure; two other representations start with a basic tree and visualize the overlapping structure by multiple trees (Carroll & Corter, 1995; Carroll & Pruzansky, 1980) or by extended trees (Corter & Tversky, 1986). Extended trees represent non-nested feature structures graphically by a generalization of the additive tree. The basic tree represents the nested structure and the non-nested structure is represented by added marked segments that cut across the clusters of the tree structure (Carroll & Corter, 1995, p. 288). The FNM is the only model that represents this overlapping feature structure by a network representation.

Imposing restrictions on the feature structure in FNM allows for other graphical representations than a network. Chapter 3 shows that an additive tree is a special subgraph of the complete feature graph, where each edge is represented by a separate feature. To obtain an additive tree representation as in Figure 1.6, the feature matrix must satisfy certain conditions. To produce a tree graph with FNM a nested set of features is not sufficient. A set of internal nodes (hypothetical stimulus objects) has to be added to the set of actual stimulus objects, or external nodes. As will become clear in Chapter 3, if the internal nodes have the correct feature structure, they will force the betweenness condition to hold in such a way that a tree graph results.

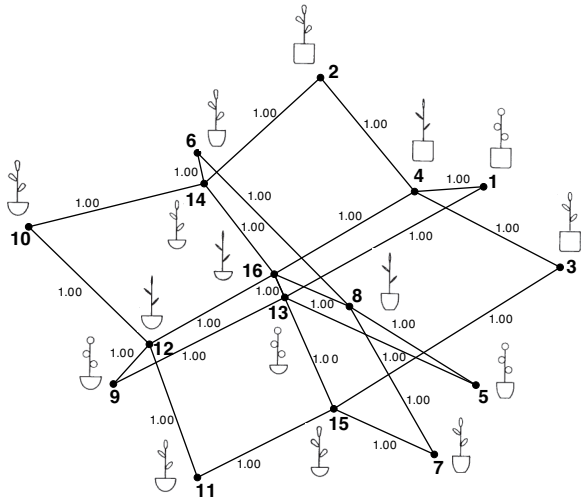

Figure 1.7: Feature network representing a 6-dimensional hypercube based on the unweighted, reduced set of features of the *plants* data. Embedding in 2-dimensional Euclidean space was achieved with PROXSCAL allowing ordinal proximity transformation with ties untied and the Torgerson start option.

Feature networks and the city-block model

The symmetric set difference is a special case of the city-block metric with binary dimensions represented by the distinctive features. Therefore, the network representation lives in city-block space. The dimensionality of this city-block space is defined by the number of features T forming a T-dimensional rectangular hyperblock, or hypercuboid with the points representing the objects located on the corners. In the special case when the symmetric set difference is equal for adjacent objects in the graph, the structure becomes a hypercube. The feature structure of the *plants* data yields a hypercube structure when all feature discriminability parameters are set equal to one. Figure 1.7 shows the resulting 6-dimensional network structure using the theoretical features and after sorting out the triangle equalities. Using the weighted feature distance transforms the lengths of the edges of the 6-dimensional hypercube into a 6-dimensional hypercuboid as in Figure 1.5. (The visual comparison between the network representations in the Figures 1.5 and 1.7 requires some effort because due to the embedding in lower dimensional space, the emplacement of the plants has changed.)

Chapter 5 demonstrates that there exists a universal network representation of city-block models. The results rely on the additivity properties of the city-block distances and the key elements of the network representation consisting of betweenness, metric segment additivity and internal nodes. The universal network construction rule also applies to other models beside the distinctive features model, namely

the common features model (additive clustering), hierarchical trees, additive trees and extended trees.

1.3 Feature Network Models: estimation and inference

Figure 1.8 shows an overview of the steps necessary to fit a Feature Network Model on data using the program PROXGRAPH that has been developed in Matlab. Starting with observed dissimilarities and a set of features, the feature discriminability parameters are estimated as well as the feature distances. The estimated feature distances lead to an adjacency matrix after being processed by the triangle equality algorithm and to coordinates obtained with PROXSCAL (Meulman & Heiser, 1999), leading to the final result, a feature network. As explained so far, the network representation of the dissimilarity data provides a convenient way to describe and display the relations between the objects. At the same time the network representation suggests a psychological model that relates mental representation to perceived dissimilarity. The psychological model is not testable with the graphical representation only. In FNM the psychological model can be tested by assessing which feature(s) contributed more than others to the approximation of the dissimilarity values. The statistical inference theory proposed in this monograph derives from the multiple regression framework, as will become clearer in the following. An important topic of this monograph is the estimation of standard errors for the feature discriminability parameters in order to construct 95% confidence intervals . Another way to decide which features are important is to use model selection techniques to obtain a relevant subset of features.

Statistical inference

The use of features, when considered as prediction variables, leads in a natural way to the univariate multiple regression model, which forms the starting point for statistical inference. It is however not the standard regression model because positivity restrictions have to be imposed on the parameters. The feature discriminability parameters represent edge lengths in a network or a tree and, by definition, networks or trees with negative edge lengths have no meaning and cannot adequately represent a psychological theory. The problem becomes more prominent in additive tree representations because each edge is represented by a separate feature. Therefore, the feature discriminability parameters are estimated by adding the following positivity constraint to Equation 1.5:

$$\min_{\boldsymbol{\eta}} = \|\boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta}\|^2 \qquad \text{subject to } \boldsymbol{\eta} \geq 0. \tag{1.7}$$

The multiple regression approach has been used earlier in models related to FNM. The Contrast Model (Takane & Sergent, 1983), the common features model (Arabie & Carroll, 1980), and the tree models (Corter, 1996) use ordinary least squares to estimate the parameters of the models. The use of nonnegative least squares is sparse in the literature of tree models. Arabie and Carroll (1980) implemented a subroutine in the MAPCLUS algorithm that encourages the weights to become positive,

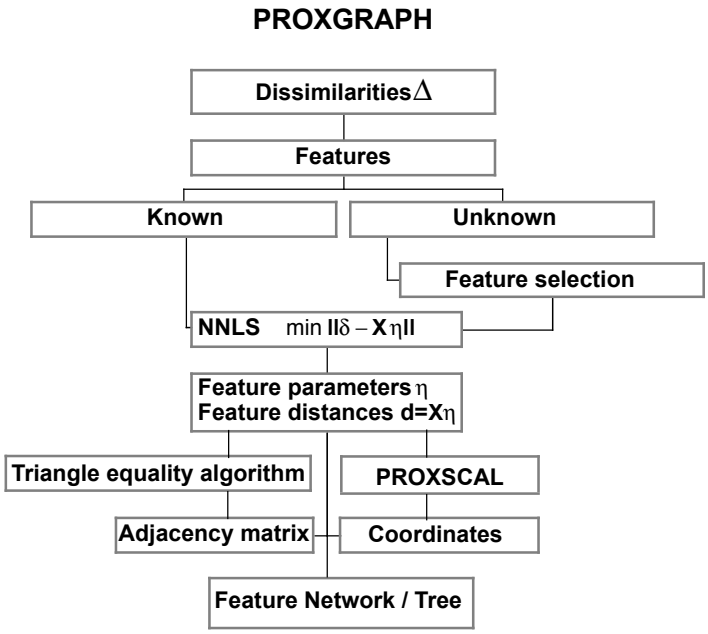

Figure 1.8: An overview of the steps necessary to fit Feature Network Models with PROXGRAPH.

but claim to explicitly avoid the use of nonnegative least squares because in the context of iterative algorithm it would reduce the numbers of clusters in the solution. Hubert, Arabie and Meulman (2001) have successfully implemented nonnegative least squares in their algorithm for the estimation of the edge lengths in additive trees and ultrametric trees. In the domain of phylogenetic trees, nonnegative least squares has been introduced by Gascuel and Levy (1996).

While nonnegative least squares has been used to estimate the parameters in models related to FNM, there are no examples in the literature of the estimation of theoretical standard errors, with or without nonnegativity constraints. The models related to FNM, the extended tree models (Corter & Tversky, 1986), the CLUSTREE models (Carroll & Corter, 1995) and, the Modified Contrast Model (Navarro and Lee, 2004) do not explicitly provide a way to test for significance of the features. It should be noted, however, that Corter and Tversky (1986) use a descriptive version of the *F*-test by permuting residuals to test the significance of the overlapping features added to the additive tree solutions. The other network models (Klauer, 1989, 1994; Klauer & Carroll, 1989) only give the best fitting network and yield no standard errors for the parameter estimates. Krackhardt (1988) provided a way to test the significance of

Table 1.3: Feature discriminability estimates, standard errors and 95% confidence intervals for *plants* data using six features selected from the complete experimental design in Table 1.1 and associated with the network graph in Figure 1.5 ($R^2 = 0.60$).

Features	$\hat{\eta}$	$\hat{\sigma}_{\hat{\eta}}$	95% t-CI	
a	6.29	0.57	5.15	7.42
b	3.64	0.57	2.50	4.77
c	2.85	0.57	1.71	3.98
d	0.00*	0.00	0.00	0.00
p	6.86	0.57	5.73	8.00
q	3.95	0.57	2.82	5.08
r	3.09	0.57	1.96	4.22
s	0.00*	0.00	0.00	0.00

* To avoid multicollinearity, the fourth level of the flowerpots (d) and the plants (s) has been omitted.

regression coefficients in networks for dyadic data that suffer from various degrees of autocorrelation by using quadratic assignment procedures. Unfortunately, his results do not apply to FNM because of the presence of constraints on the feature parameters.

Statistical inference in inequality constrained least squares problems is far from straightforward. A recent review by Sen and Silvapulle (2002) showed that topics on statistical inference problems when the associated parameters are subject to possible inequality constraints abound in the literature, but solutions are sparse. In the context of the inequality constrained least squares problem, only one author (Liew, 1976) has produced a way to compute theoretical standard errors for the parameter estimates. Liew (1976), however, did not evaluate the sampling properties of the theoretical standard errors. Chapters 2 of this monograph shows an application of the theoretical standard errors and associated 95% confidence intervals for feature networks with a priori known features. The performance of the theoretical standard errors is compared to empirical standard errors using Monte Carlo simulation techniques. Chapter 3 evaluates the performance of the theoretical standard errors for features structures in additive trees and the results are extended to the case where the feature structure (i.e., the tree topology) is not known in advance.

Table 1.3 shows the feature discriminability parameters and the associated theoretical standard errors and 95% t-confidence intervals for the theoretic features of the *plants* data. To avoid multicollinearity, the feature discriminability parameters and the associated standard errors have been estimated with a smaller feature set, than the set of theoretical features presented in Table 1.1. Two features have been omitted, namely the fourth level of the flowerpots (feature d) and the fourth level of plants (feature s). As a result these two features have zero values in Table 1.3. The overall fit of the model is reasonable with an R^2 equal to 0.60. The estimates of the feature discriminability parameters indicate that the features a, representing the square formed pot, and p, representing the round shaped leaves, are the most important in

Table 1.4: Feature matrix resulting from feature subset selection with the Positive Lasso on the *plants* data.

Plants		Features					
		F_1	F_2	F_3	F_4	F_5	F_6
1	ap	1	1	0	1	1	1
2	aq	1	1	1	0	1	1
3	ar	1	1	1	0	0	1
4	as	1	1	1	0	0	0
5	bp	0	1	0	1	1	1
6	bq	0	1	1	0	1	1
7	br	0	1	1	0	0	1
8	bs	0	1	1	0	0	0
9	cp	0	0	0	1	1	1
10	cq	0	0	1	0	1	1
11	cr	0	0	1	0	0	1
12	cs	0	0	1	0	0	0
13	dp	0	0	0	1	1	1
14	dq	0	0	0	0	1	1
15	dr	0	0	0	0	0	1
16	ds	0	0	0	0	0	0

distinguishing the plants. The network representation in Figure 1.5 reflects the importance of the features *a* and *p* by larger distances between plants that possess these features and plants that do not possess these features. The edges in the network (Figure 1.5) are labeled with the feature distances, which can be reconstructed from the feature discriminability parameters in Table 1.3. For example, the distance between the plants 2 and 4 is equal to 3.95, which is the sum of the feature discriminability parameters corresponding to their distinctive features: $q(= 3.95) + s(= 0.00)$.

Finding predictive subsets of features

In many research settings the features are not known a priori and the main objective is to find a relevant set of features that explain the dissimilarities between the objects as accurately as possible. Chapter 4 proposes a method to find adequate sets of features that is closely related to the predictor selection problem in the multiple regression framework. The basic idea is to generate a very large set of features (or, if possible, the complete set of features) using Gray codes. Since features are binary variables, they can efficiently be generated with binary coding. Next, a subset of features is selected with the Lasso option of the Least Angle Regression (LARS) algorithm (Efron, Hastie, Johnstone, & Tibshirani, 2004), a recently developed efficient model selection algorithm that is less greedy than the traditional forward selection methods used in the multiple linear regression context. To meet the positivity constraints necessary in FNM, the Lasso has been modified into a Positive Lasso. The resulting strategy incorporates model selection criteria during the search process,

leading to a set of features that is not necessarily optimal in the current data, but that constitutes a good compromise between model fit and model complexity. This approach of finding a balanced trade-off between goodness-of-fit and prediction accuracy has not been used in the psychometric models related to FNM, except for the independently developed Modified Contrast Model (Navarro & Lee, 2004) that uses a forward feature selection method and a model selection criterion related to the BIC criterion.

Table 1.4 displays the results of the Positive Lasso subset selection method on the *plants* data. The 6 selected features differ in several aspects from the theoretical features derived from the experimental design. Only the two most important features from the experimental design, features a and p were selected by the Positive Lasso (the features F_1 and F_4) in Table 1.4. Figure 1.9 represents the corresponding feature graph, which is clearly different from the feature graph based on the theoretic features (Figure 1.5): it is more parsimonious and has better overall fit ($R^2 = 0.81$). The plants have the same order in the network as in the experimental design and form a grid where each edge represents exactly one feature. For example plant number 6 and plant number 2 are connected with an edge representing the square shaped pot.

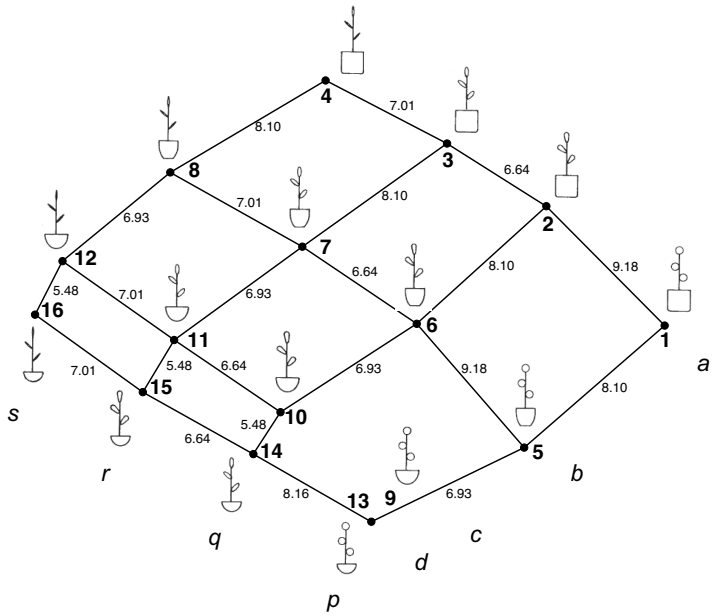

Figure 1.9: Feature graph for the *plants* data, resulting from the Positive Lasso feature subset selection algorithm on the complete set of distinctive features. The original experimental design is the cross classification of the form of the pot (a,b,c,d) and the elongation of the leaves (p,q,r,s). Embedding in 2-dimensional space was done with PROXSCAL using ratio transformation and the simplex start option. ($R^2 = 0.81$)

The edge lengths show that the pots of the form c and d are perceived as more similar (the plant numbers 9 and 13 even coincide on the same vertex in the network) than the pots with form a and b. Moreover, the network representation shows that it can perfectly represent the three types of triples of stimuli mentioned by Tversky and Gati (1982), where the geometric models based on the Euclidean metric fail. The three types of triples are:

1. *Unidimensional triple*: all stimuli differ on 1 dimension, e.g. the plants 1, 5 and 9 representing the combinations (ap, bp, cp) in Figure 1.9;
2. *2-dimensional triple*: all pairs of stimuli differ on both dimensions, e.g. the plants 1, 6 and 11 with feature combinations (ap, bq, cr);
3. *Corner triple*: two pairs differ on one dimension and one pair differs on 2 dimensions, e.g. the plants 1, 5 and 6, having feature combinations (ap, bp, bq).

Only the city-block metric or the feature network is able to correctly display the relations between these three types of triples because the triangle inequality reduces to the triangle equality in all cases. The Euclidean model and other power metrics than the city-block are able to represent unidimensional triples, and into some extent 2-dimensional triples but fail in representing the corner triples. According to the power metrics other than the city-block model, the distance between plants 1 and 6 (differing on two dimensions) is shorter than the sum of the distances between the plant pairs (1,5) and (5,6). The network representation shows that the shortest path between the plants 1 and 6 is the path from 1 to 5 to 6.

1.4 Outline of the monograph

This monograph is organized as follows. Chapter 2 explains how to obtain theoretical standard errors for the constrained feature discriminability parameters in FNM with a priori known features. The performance of the theoretical standard errors is compared to empirical standard errors (resulting from the bootstrap method) using Monte Carlo simulation techniques. Chapter 3 shows that additive trees can be considered as a special case of FNM if the objects are described by features that form a special structure. The statistical inference theory based on the multiple regression framework is further developed by extending the theory from FNM to additive trees, but also by extending the use of theoretical standard errors and associated 95% confidence intervals to a priori unknown feature structures (in this case, tree topologies). Chapter 4 proposes a new method to find predictive subsets of features, especially for the situation where the features are not known in advance. Using the multiple regression framework of the FNM, a new version of Least Angle Regression is developed that restricts the feature discriminability parameters to be nonnegative and is called the Positive Lasso. While the Chapters 2, 3 and 4 all extend the statistical inference properties of the FNM, Chapter 5 is not directly concerned with statistical inference and focuses on the properties of feature graphs. It shows that there exists a universal network representation of city-block models that can be extended to a large class of discrete models for similarity data, including the distinctive features

model, the common features model (additive clustering), hierarchical trees, additive trees, and extended trees. Chapter 6 concludes this monograph with a general conclusion and discussion.

Since the chapters 2 - 5 all represent separate papers, a certain amount of overlap is inevitable, especially in the sections describing the Feature Network Models.

Chapter 2

Estimating Standard Errors in Feature Network Models [1]

Abstract

Feature Network Models are graphical structures that represent proximity data in a discrete space while using the same formalism that is the basis of least squares methods used in multidimensional scaling. Existing methods to derive a network model from empirical data only give the best fitting network and yield no standard errors for the parameter estimates. The additivity properties of networks make it possible to consider the model as a univariate (multiple) linear regression problem with positivity restrictions on the parameters. In the present study, both theoretical and empirical standard errors are obtained for the constrained regression parameters of a network model with known features. The performance of both types of standard errors are evaluated using Monte Carlo techniques.

2.1 Introduction

In attempts to learn more about how human cognition processes stimuli, a typical psychological approach consists of analysing the ratings of perceived similarity of these stimuli. In certain situations, it is useful to characterise the objects of the experimental conditions as sets of binary variables, or features (e.g. voiced vs. unvoiced consonants). In that case it is well known that multidimensional scaling methods that embed data with underlying discrete properties in a continuous space using the Euclidean metric, will not exhaust the cognitive structure of the stimuli (Shepard, 1974, 1980, 1987). For discrete stimuli that differ in perceptually distinct dimensions like size or shape, the city-block metric achieves better results (Shepard, 1980, 1987).

In contrast to dimensional and metric methods, Tversky (1977) proposed a set-theoretical approach, where objects are characterized by subsets of discrete features.

[1] The text of this chapter represents the following article in press: Frank, L. E. & Heiser, W. J. (in press). Estimating standard errors in Feature Network Models. *British Journal of Mathematical and Statistical Psychology.* With an exception for the notes in this chapter, which are reactions to remarks made by the members of the promotion committee.

According to Tversky, the representation of an object as a collection of features parallels the mental process of participants faced with a comparison task: participants extract and compile from their data base of features a limited list of relevant features on the basis of which they perform the required task. This theory forms the basis of Tversky's Contrast Model where similarity between objects is expressed by a weighted combination of their common and distinctive features. Tversky, however, did not explain how these weights should be combined to achieve a model that could be fitted to data. Recently, Navarro and Lee (2004) proposed a modified version of the Contrast Model by introducing a new combinatorial optimisation algorithm that leads to an optimal combination of common and distinctive features.

Feature Network Models (Heiser, 1998) are a particular class of graphical structures that represent proximity data in a discrete space while using the same formalism that is the basis of least squares methods used in multidimensional scaling. Feature Network Models (FNM) use the set-theoretical approach proposed by Tversky, but are restricted to distinctive features only. It is the number of features in which two stimuli are distinct that yields a dissimilarity coefficient that is equal to the city-block metric in a space with binary coordinates, i.e., the *Hamming* distance. Additionally, the set-theoretical basis of FNM permits a representation of the stimuli as vertices in a network. Network representations are thought to be especially useful in case of nonoverlapping sets. General graphs or networks can represent parallel correspondences between the structures within two nonoverlapping subsets, which can never be achieved by continuous spatial representation nor hierarchical representations (Shepard, 1974).

In addition to the issue how to model the cognitive processing of discrete stimuli adequately, it is equally valuable to be able to decide which features are more important than others and to test which features are significantly different from zero. The models related to the FNM, the extended tree models (Corter & Tversky, 1986), the CLUSTREE models (Carroll & Corter, 1995) and, the Modified Contrast Model (Navarro & Lee, 2004) do not explicitly provide a way to test for significance of the features. The other network models (Klauer, 1989, 1994; Klauer & Carroll, 1989) only give the best fitting network and yield no standard errors for the parameter estimates.

The additivity properties of networks make it possible to consider FNM as a univariate (multiple) linear regression problem with positivity restrictions on the parameters, which forms a starting point for statistical inference. Krackhardt (1988) provided a way to test the significance of regression coefficients in networks for dyadic data that suffer from various degrees of autocorrelation by using quadratic assignment procedures. Unfortunately, his results do not apply to FNM because of the presence of constraints on the feature parameters.

Positivity restrictions on the parameters lead to an inequality constrained least squares problem. Statistical inference in inequality constrained least squares problems is far from straightforward. A recent review by Sen and Silvapulle (2002) showed that topics on statistical inference problems when the associated parameters are subject to possible inequality constraints abound in the literature. According to the authors of the review, the reason for this abundance is that optimal estimators or tests of significance generally do not exist for such nonstandard models.

In the context of the inequality constrained least squares problem, only one author (Liew, 1976) has proposed a way to compute theoretical standard errors for the parameter estimates. To the authors' knowledge there are no other examples of the application of these theoretical standard errors in the literature. Liew (1976), however, did not evaluate the sampling properties of the theoretical standard errors. The purpose of this paper is to gain more insight in the sampling distribution of the theoretical standard errors and to evaluate the usability of the standard errors in the framework of FNM in the case of known features. The accuracy of the theoretical standard errors and the use of these standard errors in constructing confidence intervals, is verified using bootstrap procedures and Monte Carlo techniques. The specific context of the FNM necessitates an adaptation of the Monte Carlo technique.

The paper is organised as follows. In the next section the Feature Network Models are described and illustrated with an application on a data set. Then, two ways of obtaining standard errors are described: theoretical standard errors and bootstrap standard errors. The results of the bootstrap study are presented in this section as well. The usability of both types of standard errors is verified by a Monte Carlo analysis, which forms the last section before the discussion.

2.2 Feature Network Models

Feature Network Models (FNM) are graphical structures that represent proximity data in a discrete space. The properties of these models will be explained using a well known data set, the perceptual confusions among 16 English consonants collected by Miller and Nicely (1955). These 16 phonemes can be described by five articulatory features: *voicing*, *nasality*, *affrication*[2], *duration*[3] and *place of articulation* (see Table 2.1). The authors were particularly interested in which articulatory features are important in distinguishing the consonants when affected by varying signal to noise conditions.

The original data consist of 17 matrices in which each cell contains the frequencies of confusion between the spoken phoneme (the rows) and the phoneme written down by the participants (the columns). Shepard (1972) converted the pooled data from the first noise condition (the first six original matrices) to a symmetric matrix of similarities with the transformation $\varsigma_{ij} = (f_{ij} + f_{ji})/(f_{ii} + f_{jj})$, where f denotes the frequencies of confusion. For our study, the similarities were further transformed into dissimilarities δ_{ij} by the transformation $\delta_{ij} = -\log(\varsigma_{ij})$, assuming that the similarity measures decay exponentially with distance.

The data are illustrative for the use of Feature Network Models because there are a priori features that describe the objects, i.e., the articulatory properties (Table 2.1). Features are binary variables indicating for each object whether a particular characteristic is present or absent. Note that features are not always intrinsically binary: any ordinal or even interval variable if categorised can be transformed into a binary feature, using dummy coding. For example, the place of articulation has three

[2] At present, phonetic experts would call this feature *friction*.

[3] The feature *duration* is not a proper phonetic feature and has been adopted arbitrarily by Miller & Nicely (1955) to distinguish the difference between {s, ʃ, z, ʒ} and the remaining consonants.

Table 2.1: Matrix of 16 English consonants, their pronunciation and phonetic features

Consonants		Features					
		$F_1{}^\star$	F_2	F_3	F_4	F_5	F_6
p	(**p**ie)	0	0	0	0	0	1
t	(**t**ie)	0	0	0	0	1	0
k	(**k**ite)	0	0	0	0	0	0
f	(**f**ie)	0	0	1	0	0	1
θ	(**th**igh)	0	0	1	0	1	0
s	(**s**igh)	0	0	1	1	1	0
ʃ	(**sh**y)	0	0	1	1	0	0
b	(**b**uy)	1	0	0	0	0	1
d	(**d**ie)	1	0	0	0	1	0
g	(**g**uy)	1	0	0	0	0	0
v	(**v**ie)	1	0	1	0	0	1
ð	(**th**y)	1	0	1	0	1	0
z	(**Z**ion)	1	0	1	1	1	0
ʒ	(vi**s**ion)	1	0	1	1	0	0
m	(**m**y)	1	1	0	0	0	1
n	(**n**ight)	1	1	0	0	1	0

$^\star F_1$ = voicing; F_2 = nasality; F_3 = affrication; F_4= duration; F_5 = place, middle; F_6 = place, front.

categories to indicate the place in the mouth where the phonemes are pronounced: front, middle and back. Dummy coding produces the two features *place, front* and *place, middle* (Table 2.1).

Feature distance

Some set theoretic properties of the binary feature matrix lead to the estimation of a distance measure that approximates the observed dissimilarities. For example, the phoneme g has one feature $\{voicing\}$ and phoneme v has the features $\{voicing, affrication, place\ front\}$. The difference between the union and the intersection (= the symmetric set difference) expresses which feature g has that v does not have and vice versa: $(g \cup v) - (g \cap v) = \{affrication, place\ front\}$. Following Goodman (1951, 1977) and Restle (1959, 1961), a distance measure that satisfies the metric axioms can be expressed as a simple count μ of the elements of the symmetric set difference between the stimuli O_i and O_j and becomes the *feature distance*: $d(O_i, O_j) = \mu[(O_i \cup O_j) - (O_i \cap O_j)]$.

Heiser (1998) demonstrated that the feature distance in terms of set operations can be re-expressed in terms of coordinates and as such, is equal to a city-block metric on a space with binary coordinates, a metric also known as the *Hamming* distance. The properties of the feature distance were known before, but it has never been used as a model to be fitted to data. If $\mathbf{E}$ is a binary matrix of order $m \times T$ that indicates which features t describe the m objects, as in Table 2.1, the re-expression of

the feature distance in terms of coordinates is as follows (Heiser, 1998):

$$\begin{aligned} d(O_i, O_j) &= \mu[(O_i \cup O_j) - (O_i \cap O_j)] \\ &= \sum_t |e_{it} - e_{jt}|, \end{aligned} \tag{2.1}$$

where $e_{it} = 1$ if feature t applies to object i, and $e_{it} = 0$ otherwise. In the example of the two phonemes g and v the feature distance is equal to 2.

For fitting purposes it is useful to generalise the distance in Equation 2.1 to a weighted count, i.e., the weighted feature distance:

$$d(O_i, O_j) = \sum_t \eta_t |e_{it} - e_{jt}|, \tag{2.2}$$

where the *feature discriminability parameters* η_t express the relative contribution of each feature.

The feature parameters are estimated by minimising the following least squares loss function:

$$\min_{\hat{\boldsymbol{\eta}}} = \|\mathbf{X}\boldsymbol{\eta} - \boldsymbol{\delta}\|^2, \tag{2.3}$$

where $\mathbf{X}$ is of size $n \times T$ and $\boldsymbol{\delta}$ is a $n \times 1$ vector of dissimilarities, with n equal to all possible pairs of m objects: $\frac{1}{2}m(m-1)$. The problem in Equation 2.3 is expressed in a more convenient multiple linear regression problem, where the matrix $\mathbf{X}$ is obtained by applying the following transformation on the rows of matrix $\mathbf{E}$ for each pair of objects, where the elements of $\mathbf{X}$ are defined by:

$$x_l = |e_{it} - e_{jt}|, \tag{2.4}$$

where the index $l = 1, \cdots, n$ varies over all pairs (i, j). The result is the binary $(0, 1)$ matrix $\mathbf{X}$, where each row represents the distinctive features for each pair of objects, with 1 meaning that the feature is distinctive for a pair of objects. The weighted sum of these distinctive features is the fitted distance for each pair of objects and is equal to $\mathbf{d} = \mathbf{X}\boldsymbol{\eta}$. Corter (1996, Appendix C, p. 57) uses a similar matrix $\mathbf{X}$ in the linear regression context to obtain the lengths of the branches in an additive tree.

The properties of the transformation in Equation 2.4 in terms of rank deficiency are not fully known yet. A full rank matrix $\mathbf{E}$ does not automatically lead to a full rank matrix $\mathbf{X}$, and a rank deficient matrix $\mathbf{E}$ does not necessarily produce a rank deficient matrix $\mathbf{X}$. In the present implementation of the Feature Network Models, this transformation is systematically checked for rank deficiency.

The feature distance parallels the path-length distance in a valued graph when one of the metric axioms, the triangle inequality, is reaching its limiting additive form $d_{ij} = d_{il} + d_{jl}$ (Flament, 1963; Heiser, 1998). Hence, sorting out the additivities in the fitted feature distances and excluding edges that are sums of other edges results in a parsimonious subgraph of the complete graph. It should be noted that the approach of sorting out the additivities is different from the network models of Klauer (1989, 1994) and Klauer and Carroll (1989), who sort out the additivities on

Table 2.2: Feature parameters, standard errors and 95% confidence intervals for consonant data

Features	$\hat{\eta}$	$\hat{\sigma}_{\hat{\eta}}$	95% CI	
Voicing	2.13	0.17	1.80	2.47
Nasality	1.32	0.22	0.88	1.76
Duration	0.98	0.19	0.60	1.36
Affrication	0.83	0.18	0.47	1.20
Place, front	0.76	0.19	0.38	1.12
Place, middle	0.57	0.18	0.21	0.93

the dissimilarities. Using the fitted distances instead leads to better networks because the distances are model quantities whereas dissimilarities are subject to error.

The feature distances ($\mathbf{d} = \mathbf{X}\boldsymbol{\eta}$) are represented as additive counts of edge lengths in the graph, where the edge lengths are the feature parameters $\boldsymbol{\eta}$. Figure 2.1 shows the network that results from the fitted distances on the consonant data. For display purposes the 6-dimensional feature network has been embedded in 3-dimensional Euclidean space by multidimensional scaling (Torgerson, 1958). Table 2.2 shows the feature discriminability parameters that result from minimising the loss function in Equation 2.3. Since the feature discriminability parameters represent edges in a network, the parameters are constrained to be nonnegative.

The values in Table 2.2 lead to the conclusion that the features *voicing* and *nasality* are the most important phonetic features used by the respondents to distinguish the 16 consonants; the phonetic features *duration* and *affrication* come in third and fourth place. The model has an R^2 equal to .90.

Feature Network Models as graphs

The network in Figure 2.1 clearly shows the distinction between the consonants based on the *voicing* feature: all voiced consonants are on the left part of the network and are well separated from the unvoiced consonants. Next, the phonetic feature of *nasality* visibly divides the two consonants *m* and *n* from the rest. The consonants *s*, ʃ, *z* and ʒ form a group in the form of rectangle and are different from the remaining 12 consonants because of the length of their pronunciation, described by the feature *duration*[4]. The most striking part of the network is the parallel structure that characterises the voiced consonants (minus the nasals) {*b*, *g*, *d*, *v*, ð, ʒ, *z*} on the one hand and the unvoiced consonants {*p*, *t*, *k*, *f*, *θ*, *s*, ʃ} on the other hand. Subsets of consonants can be distinguished by the same structure they share. For example, the voiced fricatives {*f*, *θ*, ʃ, *s*} have the same structure as the unvoiced fricatives {*v*, ð, *z*, ʒ} due to shared properties on the phonetic feature *place of articulation*.

[4]Given the arbitrarily chosen features of *duration* (see footnote 3), it would be more appropriate to state that the consonants *s*, ʃ, *z* and ʒ differ from the remaining 12 consonants in the acoustic property that is captured by the feature *duration*.

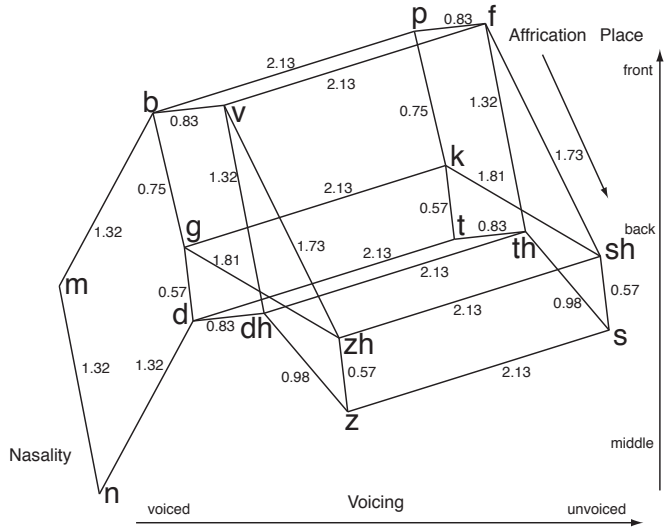

Figure 2.1: Feature Network Model on consonant data (dh = ð; zh = ʒ; th = θ; sh = ʃ).

Features: known or unknown

So far we have described the Feature Network Models in the case where features are known in advance. The example on the *consonant* data shows a typical research setting for this case, where the researchers are interested in the relative importance of specific features of the objects used in their experiment. Another research situation where the FNM could be used, is when one is primarily interested in finding the psychological features that underlie the human cognition process, and which are typically not known in advance. In this feature selection problem, the FNM use a clustering algorithm that is called cluster differences scaling[5] (Heiser, 1998).

In terms of statistical inference, the situation of known features corresponds to a univariate multiple regression problem with a fixed set of predictor variables. A different framework for statistical inference is needed for the unknown features because the predictors are random variables. The present paper addresses statistical inference with a priori features.

[5]The first application of FNM used a cluster differences scaling algorithm (Heiser, 1998) with number of clusters equal to two, which constitutes a one-dimensional MDS problem with the coordinates restricted to form a bipartition. Because it is still a hard combinatorial problem, the implementation uses a nesting of several random starts together with *K*-means type of reallocations.

2.3 Obtaining standard errors in Feature Network Models with a priori features

As explained before, the additivity properties of networks make it possible to consider Feature Network Models as a univariate multiple linear regression problem with positivity restrictions on the parameters. The constraints on the feature parameters are necessary to maintain the structural consistency of the FNM, because the feature parameters represent edge lengths in a network.

The sampling distribution of an estimator that is derived under inequality constraints is seriously affected by the constraints. In the case of nonnegativity, the sampling distribution of the least squares estimator becomes of the mixed discrete-continuous type. Without the constraints, the distribution of the least squares estimator is asymptotically normal. Imposing nonnegativity constraints causes the area of the normal density curve left of the origin to be replaced by a probability mass concentrated at the origin. Consequently, the sampling distribution of the constrained estimator is not centred around the true value anymore, and, hence the estimator is biased. This bias does not necessarily make it a worthless estimator. On the contrary, a constrained estimator will be a better estimator as the true value moves farther (in the positive direction) from the origin (*cf.* Theil, 1971)

In this context, Liew (1976) evaluated the asymptotic properties of the inequality constrained least squares estimator (ICLS) and proved that if the prior belief of positive parameters is correct, which means that it is correct to impose restrictions, the ICLS estimator is an asymptotically unbiased, consistent, and efficient estimator. In the framework of the Feature Network Models, the prior belief would be that there exists a representation of the data in terms of distances between points in a network where all edge lengths are positive. Liew (1976) also proposed a way to obtain standard errors for the ICLS estimator. The next section explains how theoretical standard errors can be obtained for the ICLS estimator.

Estimating standard errors in inequality constrained least squares

In the case that the features are known, the distinctive-feature additivity allows for considering the Feature Network Model as a univariate (multiple) linear regression model:

$$\boldsymbol{\delta} = \mathbf{X}\boldsymbol{\eta} + \boldsymbol{\epsilon} \tag{2.5}$$

where $\boldsymbol{\delta}$ is a $n \times 1$ vector with dissimilarities, $\mathbf{X}$ is a known $n \times T$ binary $(0, 1)$ matrix of rank T, and $\boldsymbol{\eta}$ is a $T \times 1$ vector. We assume that $\boldsymbol{\epsilon}$ is a $n \times 1$ random vector that follows a normal distribution,

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mu, \sigma^2\mathbf{I}), \tag{2.6}$$

where $\mathbf{I}$ is an identity matrix of rank n, and where it is assumed that σ^2 is small enough to ensure the occurrence of negative dissimilarities to be negligible. The parameters of the vector $\boldsymbol{\eta}$ are subject to positivity constraints because they represent edge lengths in the network.

The quadratic programming problem that yields the inequality constrained least squares estimator $\hat{\eta}_{\text{ICLS}}$ is the following (*cf.* Björk, 1996):

$$\begin{aligned} \min_{\hat{\boldsymbol{\eta}}} &= (\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}})'(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}) \\ &\text{subject to } \mathbf{A}\hat{\boldsymbol{\eta}} \geq \mathbf{r}. \end{aligned} \tag{2.7}$$

The matrix of constraints $\mathbf{A}$ is a $C \times T$ matrix of rank C, and $\mathbf{r}$ is a $C \times 1$ null-vector because all parameters are constrained to be greater than or equal to zero. In the case when no intercept is estimated C equals T, $\mathbf{A}$ is a $T \times T$ identity matrix, and $\mathbf{r}$ becomes a $T \times 1$ null-vector. If an intercept is estimated, there is no reason to impose restrictions on the value of this parameter because is does not directly represent an edge length in the network. In that case C equals $T - 1$.

The duality theory of the quadratic programming problem displayed in Equation 2.7 serves as the basis for the estimation of the standard errors of the parameters (Liew, 1976). The dual function of the primal problem in Equation 2.7 is:

$$\begin{aligned} &\max_{\boldsymbol{\lambda}_{\text{KT}}} = \mathbf{r}'\boldsymbol{\lambda}_{\text{KT}} + \tfrac{1}{2}(\boldsymbol{\delta}'\boldsymbol{\delta} - \hat{\boldsymbol{\eta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\eta}}), \\ &\text{subject to} \quad \mathbf{A}'\boldsymbol{\lambda}_{\text{KT}} + \mathbf{X}'\boldsymbol{\delta} = (\mathbf{X}'\mathbf{X})\hat{\boldsymbol{\eta}}, \quad \boldsymbol{\lambda}_{\text{KT}} \geq 0, \end{aligned} \tag{2.8}$$

where $\hat{\boldsymbol{\eta}}$ is a solution to the primal problem, and $\boldsymbol{\lambda}_{\text{KT}}$ is the $C \times 1$ dual vector of Kuhn-Tucker multipliers, which is the nonnegative complementary solution of the fundamental problem.

To solve the quadratic programming problem in Equation 2.7, the current implementation of PROXGRAPH uses Algorithm AS 225 (Wollan & Dykstra, 1987). This algorithm proceeds by cyclically estimating Kuhn-Tucker vectors. For the special case of nonnegative least squares, Wollan and Dykstra rephrased the problem of Equation 2.7 in a more convenient, lower dimensional space:

$$\begin{aligned} \min_{\hat{\boldsymbol{\eta}}} &= (\hat{\boldsymbol{\eta}}_{\text{OLS}} - \hat{\boldsymbol{\eta}})'\mathbf{S}^{-1}(\hat{\boldsymbol{\eta}}_{\text{OLS}} - \hat{\boldsymbol{\eta}}) \\ &\text{subject to } -\mathbf{A}\hat{\boldsymbol{\eta}} \leq 0, \end{aligned} \tag{2.9}$$

where $\hat{\boldsymbol{\eta}}_{\text{OLS}}$ is the vector with the unrestricted, ordinary least squares estimates (OLS), $\mathbf{S}^{-1}$ is equal to the inverse of $\mathbf{X}'\mathbf{X}$, and $\hat{\boldsymbol{\eta}}$ is the solution vector subject to the constraints $-\mathbf{A}\hat{\boldsymbol{\eta}} \leq 0$. The result of this optimisation problem is $\hat{\boldsymbol{\eta}}_{\text{ICLS}}$, the vector with inequality constrained least squares estimates (ICLS). Due to the different formulation of the primal problem in Equation 2.9, the resulting dual vector of Kuhn-Tucker multipliers is equal to $\frac{1}{2}\boldsymbol{\lambda}_{\text{KT}}$ in Equation 2.8.

For the solution obtained by solving Equation 2.9, the following relation exists between the ICLS estimator and the OLS estimator, using the properties of the elements of Equation 2.8 and the results obtained by Liew (1976):

$$\begin{aligned} \hat{\boldsymbol{\eta}}_{\text{ICLS}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\delta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{A}'\tfrac{1}{2}\boldsymbol{\lambda}_{\text{KT}} \\ &= \hat{\boldsymbol{\eta}}_{\text{OLS}} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{A}'\tfrac{1}{2}\boldsymbol{\lambda}_{\text{KT}}, \end{aligned} \tag{2.10}$$

where $\boldsymbol{\lambda}_{\text{KT}}$ is the vector with Kuhn-Tucker multipliers that results from solving the quadratic programming problem with Algorithm AS 225. Equation 2.10 clearly shows that if none of the elements of the vector $\hat{\boldsymbol{\eta}}_{\text{ICLS}}$ is bounded, i.e., all elements satisfy the constraint $-\mathbf{A}\hat{\boldsymbol{\eta}} \leq 0$ (Equation 2.9), then all elements of $\boldsymbol{\lambda}_{\text{KT}}$ become zero, and, as a result, the ICLS estimates reduce to the OLS estimates.

The same relation between the ICLS estimator and the OLS estimator is used to obtain standard errors for the ICLS estimates. The estimated standard errors for the OLS estimator vector are

$$\hat{\sigma}_{\text{OLS}} = \sqrt{\hat{\sigma}^2 \text{diag}\left[(\mathbf{X}'\mathbf{X})^{-1}\right]}, \tag{2.11}$$

with $\hat{\sigma}^2 = \left[(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{OLS}})'(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{OLS}})\right] / (n - T)$. The estimated standard errors for the ICLS estimator vector are

$$\hat{\sigma}_{\text{ICLS}} = \sqrt{\hat{\sigma}^2 \text{diag}\left[\mathbf{M}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{M}'\right]}, \tag{2.12}$$

where

$$\mathbf{M} = \mathbf{I} + \text{diag}[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{A}'\tfrac{1}{2}\boldsymbol{\lambda}_{\text{KT}}][\text{diag}(\hat{\boldsymbol{\eta}}_{\text{OLS}})]^{-1}. \tag{2.13}$$

If the model is unconstrained, the estimated variance-covariance matrix reduces to the variance-covariance matrix of the OLS estimator.

Determining the standard errors by the bootstrap

Considering Feature Network Models as multiple linear regression models also offers a context for the bootstrap. The bootstrap (Efron & Tibshirani, 1998) is a computer-intensive resampling method that uses the empirical distribution of a statistic to asses its variability, and is widely used as an alternative to parametric approaches.

There are two methods of bootstrapping a regression model: bootstrapping pairs and bootstrapping residuals (Efron & Tibshirani, 1998). In simple regression with one dependent variable and one predictor variable, bootstrapping pairs or bivariate sampling, implies that for each sampled observation the corresponding value of the predictor variable is sampled as well. Applied to the multiple regression situation of the Feature Network Models, bivariate sampling becomes *multivariate* sampling because there are several features, or predictor variables. The multivariate bootstrap proceeds in the following way: for each sampled observation δ_l $(l = 1, \cdots n)$, from the vector of dissimilarities of the original sample, the corresponding row $(\mathbf{x}'_l)$ of the feature matrix $\mathbf{X}$ is sampled as well. A bootstrap sample $\mathbf{b}^*_b$ $(b = 1, \cdots, B)$ taken from an original sample of n observations has the following form:

$$\mathbf{b}^*_b = \{(\boldsymbol{\delta}_l, \mathbf{x}'_l)_1, (\boldsymbol{\delta}_l, \mathbf{x}'_l)_2, \cdots, (\boldsymbol{\delta}_l, \mathbf{x}'_l)_n\}. \tag{2.14}$$

The other bootstrap method, bootstrapping residuals, does not sample directly from the observations on the dependent variable and the predictor variable, but samples with replacement from the estimated residuals obtained from fitting the regression model to the data. Fitting the Feature Network Model leads to

$$\hat{\boldsymbol{\delta}} = \mathbf{X}\hat{\boldsymbol{\eta}} + \hat{\boldsymbol{\epsilon}}, \tag{2.15}$$

where $\hat{\boldsymbol{\delta}}$ is the vector with predicted values of the dissimilarities, $\mathbf{X}$ is the fixed feature matrix, $\hat{\boldsymbol{\eta}}$ are the estimated feature parameters, and $\hat{\boldsymbol{\epsilon}}$ is the vector with estimated residuals. A bootstrap sample $\tilde{\mathbf{b}}_b$, using the method of sampling residuals, is obtained by keeping $\mathbf{X}\hat{\boldsymbol{\eta}}$ fixed and sampling with replacement from $\hat{\boldsymbol{\epsilon}}$:

$$\tilde{\mathbf{b}}_b = \{(\mathbf{x'}_1\hat{\boldsymbol{\eta}} + \hat{\epsilon}_1, \mathbf{x'}_1), (\mathbf{x'}_2\hat{\boldsymbol{\eta}} + \hat{\epsilon}_2, \mathbf{x'}_2), \cdots, (\mathbf{x'}_n\hat{\boldsymbol{\eta}} + \hat{\epsilon}_n, \mathbf{x'}_n)\}. \tag{2.16}$$

In deciding which method is better, Efron and Tibshirani (1998) argue that the choice depends on how far the linear regression model can be trusted. The linear regression model in Equation 2.5 says that the error between δ_l and its mean $\mathbf{x'}_l\boldsymbol{\eta}$ does not depend on $\mathbf{x'}_l$, which is a strong assumption that can fail even when the linear regression model is correct. Bootstrapping residuals is therefore more sensitive to assumptions than bootstrapping pairs that only assumes that the original pairs $(\delta_l, \mathbf{x'}_l)$ are randomly sampled from some distribution g. However, Efron and Tibshirani (1998) conclude that both sampling methods yield reasonable standard errors, even if the statements in Equations 2.5 and 2.6 are completely wrong.

Two arguments have lead to the choice of multivariate sampling in this study. First, the properties of the error distribution related to proximities are not sufficiently known to justify strong assumptions. The second one is a more practical argument: it is obvious from Equation 2.16 that the method of sampling residuals can lead to the undesired situation of negative dissimilarities, when by chance a large negative residual $\hat{\epsilon}_l$ is associated with a smaller value of $\mathbf{x'}_l\hat{\boldsymbol{\eta}}$.

Opposed to bivariate or multivariate sampling, where the sampling of the predictor variables (the features) depends on the sampling of the dependent variable (the dissimilarity), another approach would be to sample the predictor variables and the dependent variable independently, which is called univariate sampling (Lee & Rodgers, 1998). These authors demonstrate that bivariate sampling matches the logic of computing standard errors and constructing confidence intervals , whereas univariate sampling is more suited for hypothesis testing. The difference follows from the way the empirical sampling distribution is used to test the null hypothesis of a statistic. In univariate sampling the scores on the predictor variables are randomly matched with the scores on the dependent variable, and consequently, the expected value of the statistic is 0. The consequences for the empirical distribution resulting from the different methods is that for bivariate or multivariate sampling the empirical sampling distribution is centered around the value of the observed sample statistic and that for univariate sampling the empirical sampling distribution is centred around the value 0. Hence, in bivariate sampling $\mathbf{H}_0$ would be rejected if the middle 95% of the empirical distribution does not include the value 0 and in univariate sampling $\mathbf{H}_0$ would be rejected if the middle 95% of the distribution does not include the observed sample statistic. In this paper we are interested in obtaining standard errors and confidence intervals for the feature parameters and we are not primarily interested in hypothesis testing. Therefore, the method of choice is multivariate sampling.

Bootstrap procedures

A number of $B = 10,000$ bootstrap samples was taken from the *consonant* data (Miller & Nicely, 1955). Bootstrap samples were taken using multivariate sampling, which means that for each dissimilarity δ_l sampled from the *consonant* data, the corresponding row of the original $\mathbf{X}$ matrix with features was sampled as well. All computations were programmed with Matlab and random samples were taken using the pseudo-random number generator of Matlab, which was set to 1.0 before running the program.

Nominal standard errors, $\hat{\sigma}_{\text{OLS}}$ and $\hat{\sigma}_{\text{ICLS}}$, where estimated for the OLS and ICLS estimators (using Equations 2.11 and 2.12), as well as estimates of bias (the mean of the bootstrap replications of ICLS and OLS minus the respective sample estimates) and bootstrap standard errors sd_B (the standard deviation of the B bootstrap replications). Nominal confidence intervals , based on the t distribution ($df = n - T$, with n equal to the number of dissimilarities and T equal to the number of features), were computed for the $\hat{\eta}_{\text{OLS}}$ and $\hat{\eta}_{\text{ICLS}}$ estimators, using $\hat{\sigma}_{\text{OLS}}$ and $\hat{\sigma}_{\text{ICLS}}$. Two types of bootstrap confidence intervals were computed on the 10,000 bootstrap samples: the bootstrap-t interval and the *bias-corrected and accelerated* bootstrap interval, the BC_a (Efron & Tibshirani, 1998). The bootstrap-t interval is computed in the same way as the nominal confidence interval , with the only difference that the bootstrap standard errors are used instead of the estimated standard errors for the sample.

Nominal confidence intervals and bootstrap-t intervals are by definition symmetric, whereas BC_a intervals are only symmetric if the distribution of the statistic is symmetric, otherwise they adjust to the shape of the sampling distribution, especially in case of skewness. The BC_a follows the shape of the sampling distribution by modifying the endpoints of the interval, which are based on percentile points. This adjustment involves an extra step in the bootstrap procedure where the acceleration parameters are computed with a jackknife procedure (for details on the computations see Efron & Tibshirani, 1998, Chapter 14, and for computation in Matlab, see Martinez & Martinez, 2002, Chapter 7.4 and Appendix D.1).

Results bootstrap

Table 2.3 shows that the nominal standard errors for both $\hat{\eta}_{\text{OLS}}$ and $\hat{\eta}_{\text{ICLS}}$ estimators are almost equal to the empirical variability of these parameters captured by the bootstrap standard deviations (see columns $\hat{\sigma}_{\text{OLS}}$, $\hat{\sigma}_{\text{ICLS}}$, and sd_B). For the feature *duration*, the nominal standard error of the ICLS estimate is slightly larger than the bootstrap standard deviation. The lower value of the bootstrap standard deviations can be explained by the fact that during the sampling process the constraints are activated more often for parameter values that are almost equal to zero, and, as a result, there is less variability. In that case, the nominal standard errors overestimate the variability.

In terms of bias the OLS estimates have lower bias than the ICLS estimates (see Table 2.3). This difference is to be expected because the ICLS estimator is biased in a finite sampling situation as its empirical distribution is not centred around the true parameter value due to imposing constraints. Comparing the results in Table 2.3 to

Table 2.3: Three types of 95% Confidence Intervals for ICLS and OLS estimators resulting from the bootstrap study on the *consonant* data.

	Results ICLS *estimates*									
Features	$\hat{\eta}_{\text{ICLS}}$	Bias	$\hat{\sigma}_{\text{ICLS}}$	sd_B^a	Nominal CI^b		Boot. t CI^b		BC_a CI^b	
Constant	2.23	-0.02	0.13	0.13	1.97	2.48	1.97	2.48	1.98	2.47
Voicing	1.21	0.01	0.11	0.11	0.99	1.42	0.99	1.42	0.99	1.42
Nasality	0.78	-0.00	0.13	0.12	0.52	1.03	0.53	1.02	0.53	1.01
Affrication	0.37	-0.00	0.11	0.11	0.14	0.59	0.16	0.58	0.16	0.58
Duration	0.09	0.01	0.12	0.09	-0.15	0.33	-0.09	0.27	0.00	0.31
Place, middle	0.08	0.02	0.07	0.09	-0.06	0.23	-0.09	0.26	0.00	0.29
Place, front	0.00	0.00	0.00	0.01	0.00	0.00	-0.02	0.02	0.00	0.07
	Results OLS *estimates*									
Features	$\hat{\eta}_{\text{OLS}}$	Bias	$\hat{\sigma}_{\text{OLS}}$	sd_B^a	Nominal CI^b		Boot. t CI^b		BC_a CI^b	
Constant	2.34	-0.00	0.13	0.14	2.07	2.60	2.06	2.61	2.05	2.59
Voicing	1.19	0.00	0.11	0.11	0.98	1.40	0.98	1.41	0.98	1.41
Nasality	0.77	0.00	0.13	0.12	0.52	1.02	0.53	1.01	0.53	1.00
Affrication	0.36	0.00	0.11	0.10	0.14	0.58	0.15	0.57	0.16	0.57
Place, middle	0.13	-0.00	0.11	0.11	0.09	0.34	-0.09	0.35	-0.10	0.33
Duration	0.08	0.00	0.11	0.11	-0.13	0.30	-0.13	0.29	-0.11	0.30
Place, front	-0.22	0.00	0.11	0.11	-0.43	0.00	-0.43	-0.00	-0.42	-0.01

[a] Standard deviation based on $B = 10,000$ bootstrap samples.

[b] For each confidence interval (CI) the left column corresponds to the lower end point of the interval and the right column to the upper end point.

the empirical distribution of the ICLS and OLS estimates in Figure 2.2, leads to the following conclusions. The higher values of bias occur for the features *duration* and *place middle*, where the constraints are activated more often because these features have parameter values almost equal to zero. The irregularity in the activation of the constraints, i.e., sometimes they are activated and sometimes not, leads to an empirical distribution that is not centred around the true value. In contrast, the feature *place front* has almost no bias because the constraints are activated almost all the time, resulting in a distribution centred around zero, which is the true value. Even if bias is present, it is not substantial when compared to the bootstrap standard deviations: in all cases the ratio of the bias divided by the standard deviation is lower than .25, a critical value for bias proposed by Efron and Tibshirani (1998).

A way to evaluate the performance of the nominal standard errors of the ICLS estimator, is to compare the nominal confidence intervals of this estimator with the nominal confidence intervals of the OLS estimator. Table 2.3 shows that, in general, the nominal confidence intervals for both the OLS and ICLS estimators follow the

empirical confidence intervals (standard bootstrap and BC_a) very closely, except for the three ICLS estimators of the features *duration*, *place middle* and *place front*, where constraints are activated. Figure 2.3 clearly displays the difference between the standard bootstrap interval and the nominal interval on the one hand, and the difference between the BC_a interval and the nominal interval, on the other hand. Figure 2.3 also illustrates how the BC_a interval results in adjustments of both endpoints of the interval, in an attempt to approximate the shape of the empirical distribution. Figure 2.4 displays the comparison between the ICLS estimator and the OLS estimator, and also includes the BC_a intervals, which give the best available estimation of the parameter space. Figure 2.4 leads to the conclusion that for the features where constraints are activated, sometimes the nominal confidence intervals tend to be slightly larger than the empirical confidence intervals .

To answer the question whether the feature parameter values are significantly different from zero, the three types of confidence intervals for both the OLS and ICLS estimators are unanimous within each estimator, but lead to a slightly different conclusion for the separate estimators. In case of the ICLS estimator, the parameters *duration*, *place middle* and *place front* are not significantly different from zero, and, in case of the OLS estimator only *place middle* and *duration* are not significantly different from zero. In conclusion, the nominal standard errors for the ICLS estimator perform equally well as the nominal standard errors for the OLS estimator, even if the ICLS estimator is slightly biased.

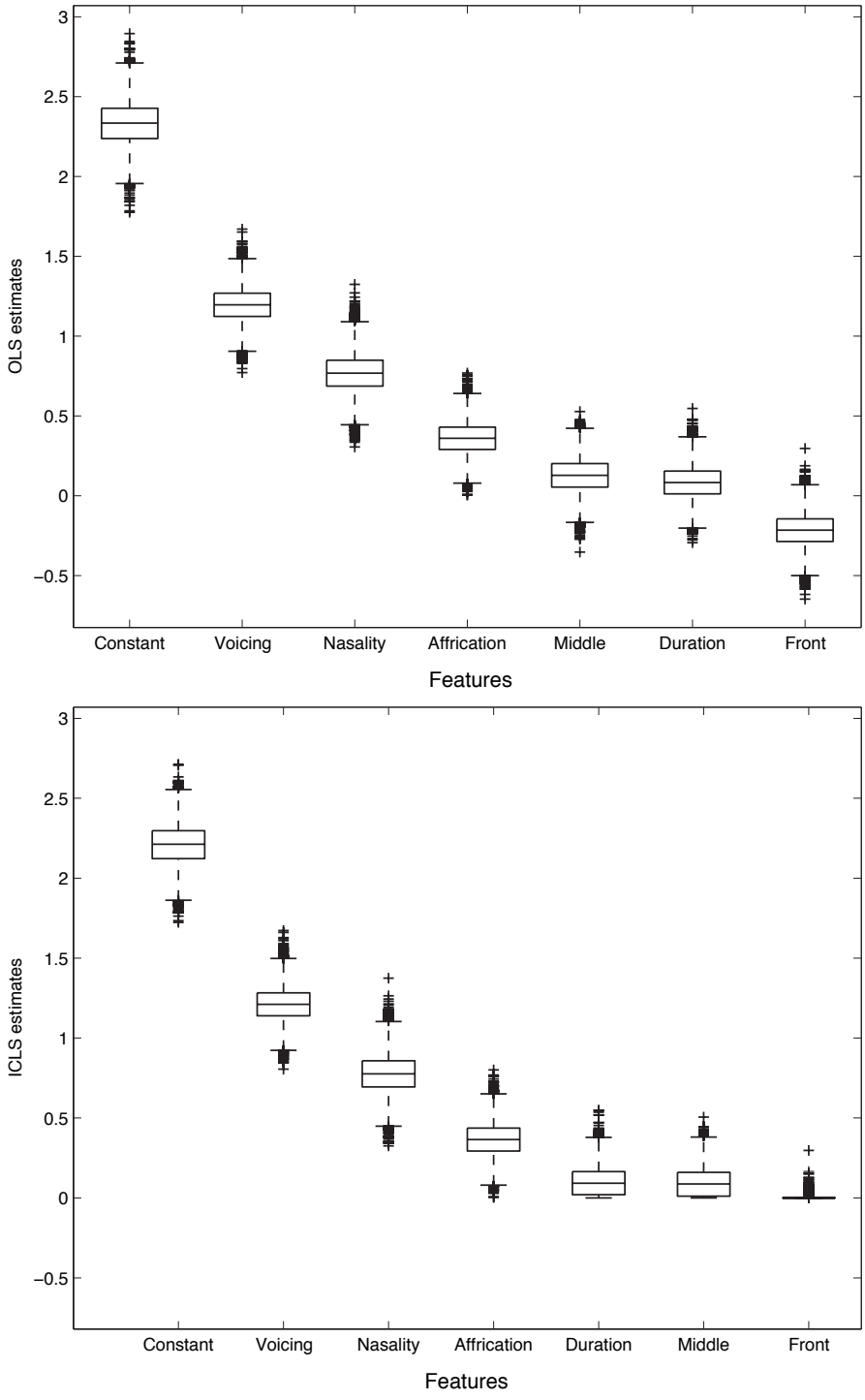

Figure 2.2: Empirical distribution of OLS (top) and ICLS (bottom) estimators (1,0000 bootstrap samples).

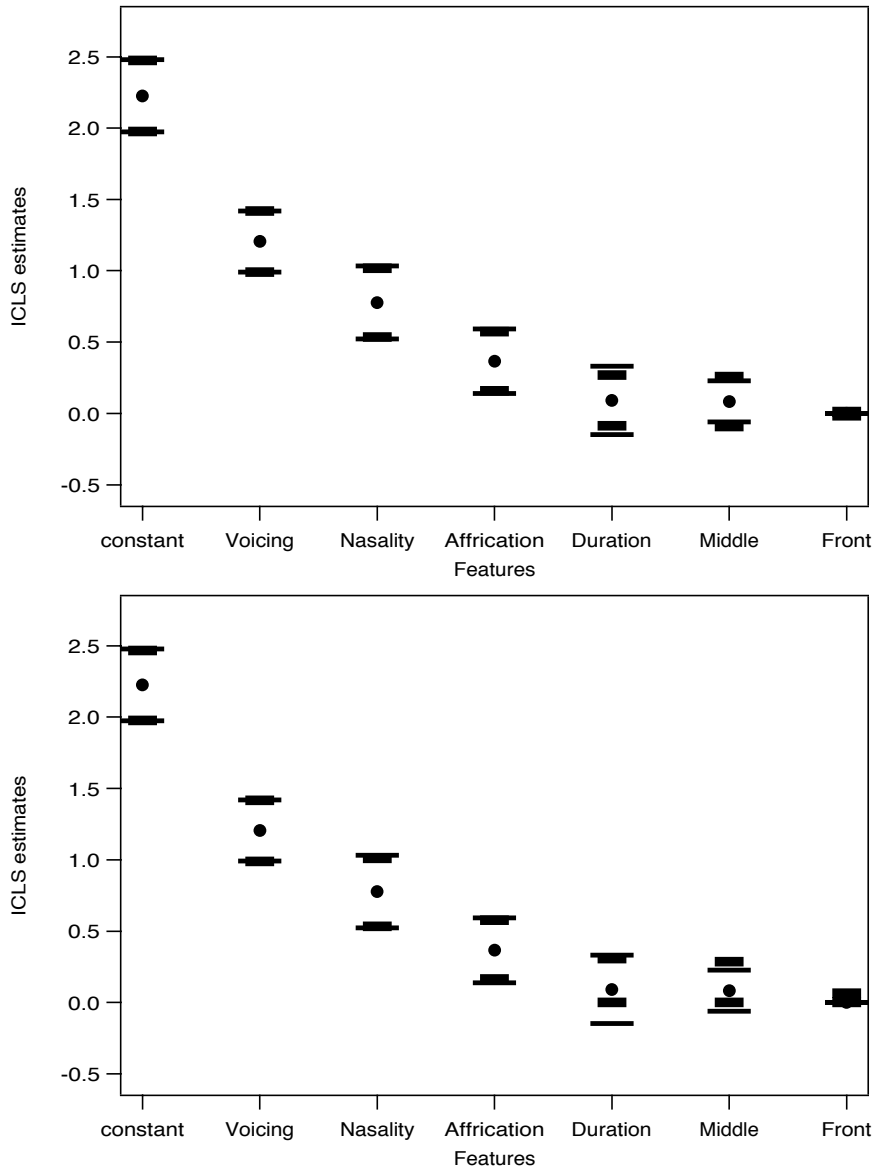

Figure 2.3: Comparison of nominal confidence intervals for ICLS estimator with bootstrap-t CI (top) and bootstrap BC_a CI (bottom); long bar = nominal CI; short bar = bootstrap-t CI or BC_a CI.

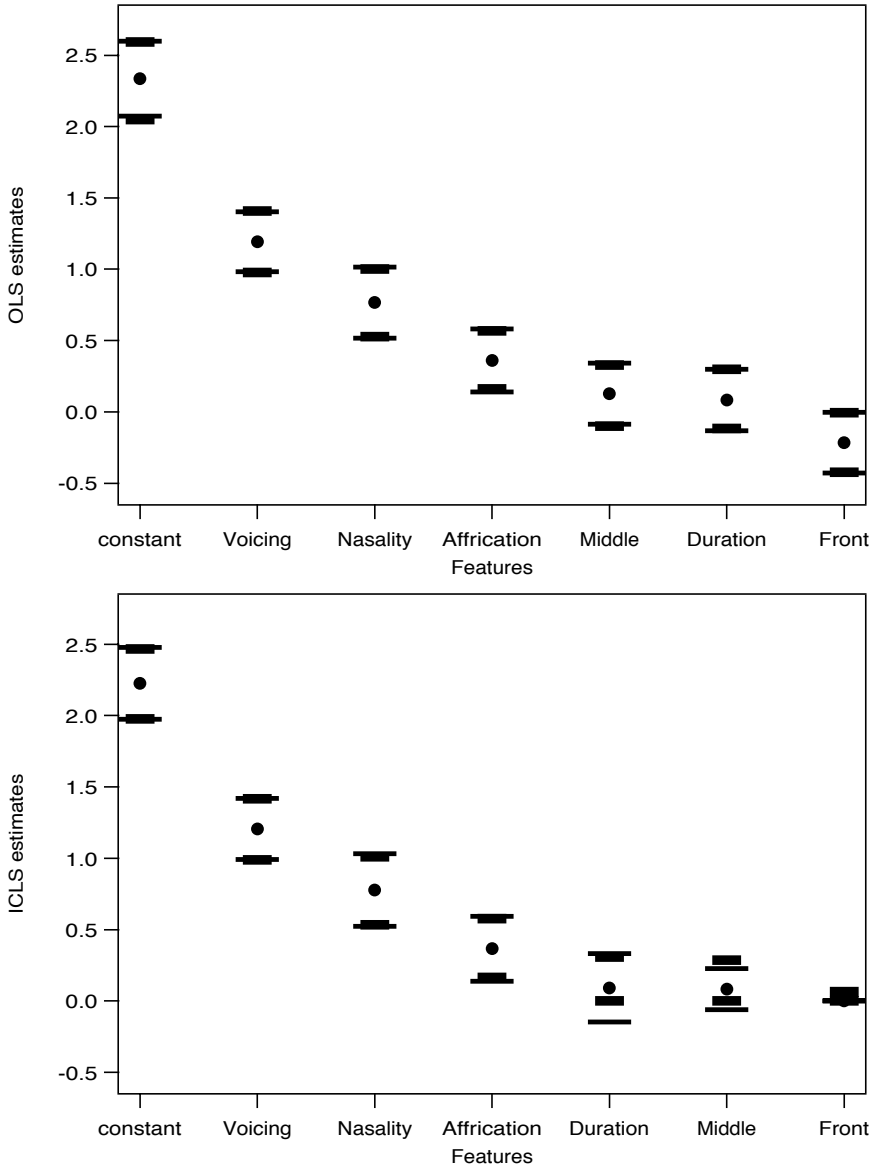

Figure 2.4: BC_a and nominal confidence intervals for OLS and ICLS estimators (long bar = nominal CI; short bar = BC_a CI).

2.4 Monte Carlo simulation

The purpose of the simulation study is to evaluate the performance of the nominal standard errors of the ICLS estimator compared to empirical (bootstrap) standard errors. In addition, the performance of these nominal standard errors are evaluated by comparing the coverage of the nominal confidence intervals with the coverage of bootstrap confidence intervals . The coverage is equal to the proportion of times the true value is included in the confidence interval.

The performance of the standard errors of the ICLS estimator was evaluated using positive true feature parameters, which represents a situation where it is correct to apply constraints and consequently, the asymptotic properties of the ICLS estimator are expected to hold. For the asymptotic properties to hold, normally distributed errors and homogeneous variances are required as well. Given positive true feature parameters, true distances can be computed that can be used as population values from which dissimilarities can be sampled by adding some error to the true distances.

However, sampling dissimilarities that meet the properties of the normal distribution and homogeneous variances is not straightforward. A way to obtain dissimilarities that is commonly used in the multidimensional scaling context is the following (see for example, Weinberg, Carroll, & Cohen, 1984): first, one computes true distances on some a priori determined coordinates. Next, one adds disturbances by multiplying the distances by $\exp(\hat{\sigma} \times z)$, where $\hat{\sigma}$ is the sample standard deviation obtained from a real data set, and z is an independently sampled standard normal deviate. The resulting dissimilarities are lognormally distributed with location parameter d and dispersion $\hat{\sigma}$. Lognormally distributed dissimilarities are not suitable for the current situation because we use the standard least squares framework with normal errors. Therefore, we created a method that allows for sampling dissimilarities with the required properties of normality and homogeneous variances. The new method uses the binomial distribution, as will be explained in the next section.

Sampling dissimilarities from the binomial distribution

If Y is a binomially distributed random variable, $Y \sim \mathcal{B}in(\kappa, p)$, then it is well known that the expected value of Y is $E(Y) = \kappa p$ and the variance of Y is $Var(Y) = \kappa p(1 - p)$. If N independent random variables are binomially distributed, $Y_\ell \cdots Y_N \sim \mathcal{B}in(\kappa, p)$, then the expected value of the *mean* of the N random variables equals

$$E(\overline{Y}) = \frac{1}{N} \sum_{\ell=1}^{N} E(Y_\ell) = \kappa p = \mu, \tag{2.17}$$

and the variance of the mean is equal to

$$Var(\overline{Y}) = \frac{1}{N^2} \sum_{\ell=1}^{N} Var(Y_\ell) = \frac{\kappa p(1-p)}{N} = \frac{\sigma^2}{N}. \tag{2.18}$$

If N is large enough, the distribution of the mean of N binomially distributed variables will approximate the normal distribution with the following parameters:

$$\overline{Y} \sim \mathcal{N}(\mu, \frac{\sigma^2}{N}). \tag{2.19}$$

The binomial distribution offers the possibility to sample dissimilarities within the framework of the normal distribution. The dissimilarities can be viewed as resulting from a process where N participants evaluate the degree of dissimilarity of $n = \frac{1}{2}m(m-1)$ object pairs on an κ-points scale, where a large number means that a pair of objects is very dissimilar. The result is an $n \times N$ matrix $\widetilde{\mathbf{\Delta}}$ of random variables with range $[0, \kappa]$. The elements of $\widetilde{\mathbf{\Delta}}$ are denoted by $\widetilde{\Delta}_{l\ell}$ ($l = 1, 2, \cdots, n$; $\ell = 1, 2, \cdots, N$).

All elements in some row of $\widetilde{\mathbf{\Delta}}$ follow a binomial distribution with κ equal to the total number of points on the scale, and p_l the binomial parameter. When two objects are very dissimilar, the value of p_l will be larger because more participants will evaluate the resemblance of the objects with larger κ-values. The expected value of the mean $\overline{\Delta}_l$ of each row is

$$E(\overline{\Delta}_l) = E\Big[\frac{1}{N}\sum_{\ell=1}^{N}\widetilde{\Delta}_{l\ell}\Big] = \frac{1}{N}\sum_{\ell=1}^{N}E(\widetilde{\Delta}_{l\ell}) = \kappa p_l = d_l, \tag{2.20}$$

where d_l is the true distance for object pair l. The variance of $\overline{\Delta}_l$ is

$$Var(\overline{\Delta}_l) = \frac{1}{N^2}\sum_{\ell=1}^{N}Var(\widetilde{\Delta}_{l\ell}) = \frac{d_l(1-p_l)}{N}. \tag{2.21}$$

If the number of replications N is large enough, the distribution of the mean $\overline{\Delta}_l$ approximates the normal distribution with the following parameters:

$$\overline{\Delta}_l \sim \mathcal{N}\Big(d_l, \frac{d_l(1-p_l)}{N}\Big). \tag{2.22}$$

From this set-up, it follows that the random variables $\widetilde{\Delta}_{l\ell}$ are identically distributed with expected value d_l. Let $\tilde{\delta}_{l\ell}$ denote a realisation from $\widetilde{\Delta}_{l\ell}$. The sampling process follows the steps in Figure 2.5. The first step is to sample N replications from a binomial distribution with p_l equal to d_l/κ. The result is a matrix of size $n \times N$ with binomial scores $\tilde{\delta}_{l\ell}$. Each of the n simulated dissimilarity values is obtained by taking the mean of each row of this matrix, which is equal to:

$$\delta_l = \frac{1}{N}\sum_{\ell=1}^{N}\tilde{\delta}_{l\ell}, \tag{2.23}$$

and the resulting dissimilarities approximate the normal distribution shown in Equation 2.22.

During the sampling process the variance of the dissimilarities can be manipulated because the magnitude of the variance depends on the number of replications

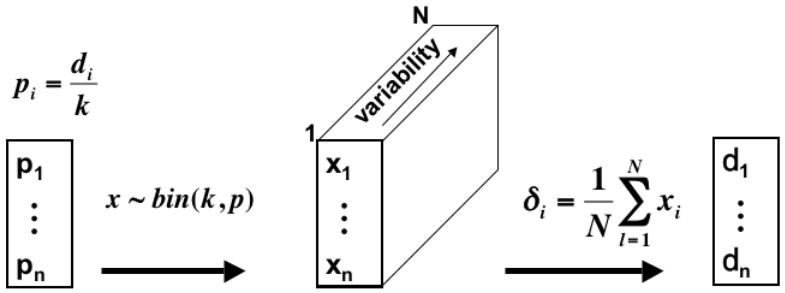

Figure 2.5: Sampling dissimilarities from a binomial distribution

N. A large number of replications leads to lower variance levels, and a small number to higher variance levels. Figure 2.5 displays a situation of heterogeneous variance because each row of the matrix has the same number of replications N, but a different value of σ^2 due to different values of p_l. The situation of homogeneous variances can be obtained by choosing the value of N for each row in such a way that the resulting variance is equal for each row. Given a situation of homogeneous variance, one can obtain a heterogeneous variance condition by choosing N equal to the mean of the N values needed for the homogeneous variance situation. The result is a vector of heterogeneous variances that are centered around the value of the variance of the homogeneous variance

Simulation procedures

The simulation proceeded as follows. True distances were computed with:

$$\mathbf{d} = \mathbf{X}\boldsymbol{\eta}, \tag{2.24}$$

where the true parameters are equal to the ICLS estimates ($\hat{\eta}_{\text{ICLS}}$) in Table 2.3 and $\mathbf{X}$ is obtained with the feature matrix of the consonant data (Table 2.1). A number of $S = 1,000$ samples of $n = 120$ dissimilarities each, was created by sampling from the binomial distribution as described before, with p_l equal to d_l/κ, where the d_l are the distances from Equation 2.24, and κ equals 15. A homogeneous variance condition was created with σ^2 equal to 0.34, which corresponds to the observed residual error variance after fitting the Feature Network Model on the consonant data.

Each simulation sample formed the starting point for a bootstrap of $B = 10,000$ samples, using the method of multivariate sampling. The simulation procedures were programmed in Matlab and made use of its pseudo-random number generator, which was set to 1.0 prior to the simulation process.

The simulation (based on $S = 1,000$ samples) yielded 1,000 nominal standard errors ($\hat{\sigma}_{\text{ICLS}}$, $\hat{\sigma}_{\text{OLS}}$) for the ICLS and OLS estimators. The 1,000 bootstraps (each based on $B = 10,000$ bootstrap samples) resulted in 1,000 bootstrap standard deviations (sd_B) of the ICLS and OLS estimators.

The bias and the root mean squared error (*rmse*) are commonly used measures to evaluate the performance of estimates (*cf.* Efron & Tibshirani, 1998; Freedman & Peters, 1984). Good estimators are unbiased and have small *rmse*. The estimation of bias is equal to the expected value of a statistic, $E(\hat{\theta})$, minus the true value θ. Relative bias estimates, which are equal to $[E(\hat{\theta}) - \theta]/\theta$, are useful for comparisons between parameter values of different magnitude. The *rmse* is equal to the square root of $E[(\hat{\theta} - \theta)^2]$ and takes into account both bias and standard error of an estimate, as can be deduced from the following decomposition (Efron & Tibshirani, 1998):

$$rmse = \sqrt{sd_{\hat{\theta}}^2 + bias_{\hat{\theta}}^2}. \tag{2.25}$$

Estimates of bias were calculated for the feature parameter estimates $\hat{\eta}_{\text{ICLS}}$, the nominal standard errors ($\hat{\sigma}_{\text{ICLS}}$, $\hat{\sigma}_{\text{OLS}}$), and the bootstrap standard deviations. For example, the bias of each nominal standard error $\hat{\sigma}_\eta$ is estimated by:

$$bias_{\hat{\sigma}_\eta} = \left[\frac{1}{S}\sum_{a=1}^{S}\hat{\sigma}_{\eta_a}\right] - \sigma_\eta, \tag{2.26}$$

where S indicates the number of simulation samples, and η stands for the ICLS or the OLS estimator. The bias of $\hat{\sigma}_{\text{OLS}}$ is calculated using Equation 2.11, with the difference that σ^2 and $\mathbf{X}$ are the true standard deviation and true predictors used to create the simulation samples, as explained in the beginning of this section. The bias of $\hat{\sigma}_{\text{ICLS}}$ is computed with Equation 2.12, using the true values σ^2, $\mathbf{X}$ and $\mathbf{M}$ from Equation 2.13. The bias for the bootstrap standard errors is calculated in the same way, with the exception that $\frac{1}{S}\sum_{a=1}^{S}\hat{\sigma}_{\eta_a}$ is replaced by the sum of the bootstrap standard deviations sd_B.

The nominal standard errors were used for the construction of nominal 95% confidence intervals. Empirical 95% confidence intervals were calculated as well, using the same intervals as in the bootstrap study, i.e., the bootstrap-*t* confidence interval and the BC_a confidence interval . The performance of all confidence intervals was evaluated by computing coverage percentages. The coverage percentage is equal to the proportion of the simulated samples in which the confidence interval includes the true parameter value. The presence of a true feature parameter equal to zero allows for calculating the empirical alpha, which is the proportion of times the interval contains a zero and leads to the incorrect rejection (given the true value equal to zero) of $\mathbf{H}_0$ (*cf.* Lee & Rodgers, 1998). Following the same logic, the other, nonzero feature parameters, are suitable for the calculation of the empirical power by counting the number of times the interval contains a zero, which leads to the correct rejection of the $\mathbf{H}_0$.

Additional simulation studies

The same simulation procedures described in the previous section were repeated using the structures derived from three additional data sets. The data sets were selected on the presence of a clear feature structure of the stimuli that the authors intended to test in their experiments. Besides the number of stimuli (objects) that

varies from 9 to 36 in the data sets, another important characteristic of the data is the different numbers of true parameter values that are equal or close to zero. True parameter values that approach zero lead to an increasing number of activated constraints during the simulation process, which will give a better insight in the properties of the nominal and the empirical standard errors of the constrained least squares estimator.

The first data set is the *similarity of faces* data (Corter & Tversky, 1986) where the stimuli consist of 9 schematic faces constructed factorially using three different shapes (Top-Heavy, Even, Bottom-Heavy) and three different expressions (Smile, Neutral, Frown). The participants were asked to rate the similarity of the faces between all pairs of faces on a 9-point scale. The feature structure found by the authors is presented in the first part of Table 2.4. Fitting the Feature Network Model using this feature structure yields an R^2 of 99.73 and the feature parameter values that are shown in Table 2.4. From these feature parameter values true distances were derived using $\kappa = 9$ (based on the 9-point scale used in the experiment) and, an error variance equal to 0.03, which corresponds to the observed residual error variance after fitting the Feature Network Model on the *similarity of faces* data. The second data set is the *Swedish letters* data (Kuennapas & Janson, 1969), where 57 participants judged the similarity of all unique pairs of the 28 Swedish letters on a 100-point scale. Table 2.4 presents the feature structure that the authors obtained from a factor solution excluding loadings < 0.30. The fit of the FNM on this feature structure leads to an R^2 of 96.51 and the feature parameters that are displayed in Table 2.4. The true distances used for the simulation were derived from these feature parameters with $k = 100$ as in the experiment, and an error variance of 0.02, based on the original sample. The third data set is the well known *Morse code* data by (Rothkopf, 1957), which concerns the ratings of all possible pairs of the 36 Morse codes by 150 participants who did not know the code. We used the 2-dimensional MDS solution by Shepard (1980) to derive the feature structure shown in Table 2.4. The feature parameter values resulting from fitting the feature structure wth FNM is presented in Table 2.4 and the R^2 equals 92.70 with a residual error variance of 0.15. This variance value together with a κ equal to 100 were used to derive true distances for the simulation study.

Table 2.4: Description of features and the corresponding objects for three additional data sets

Feature	Description	Objects	$\hat{\eta}_{\text{ICLS}}$
Features for similarity of faces data, based on the extended tree solution (Corter & Tversky, 1986)			
F_0	Universal feature	All objects	1.54
F_1	Top-Heavy (T)	TS, TN, TF	0.51
F_2	Even (E)	ES, EN, EF	0.62
F_3	Bottom-Heavy (B)	BS, BN, BF	0.00
F_4	Smile (S)	TS, ES, BS	0.38
F_5	Neutral (N)	TN, EN, BN	0.81
F_6	Frown (F)	TF, EF, BF	0.70
Features for Swedish letters data based on the factor solution with loadings $\geqslant$ 0.30 (Kuennapas & Janson, 1969)			
F_0	Universal feature (intercept)	all 28 letters	0.53
F_1	Vertical linearity	t, f, l, r, i, j	0.13
F_2	Roundness	o, c, ö, e	0.04
F_3	Parallel vertical linearity	n, m, h, u, r	0.05
F_4	Vertical linearity with dot	i, j, l	0.00
F_5	Roundness attached to vertical linearity	q, p, g, b, d, o, h, y	0.06
F_6	Vertical linearity with crossness	k, h, b, x, d	0.00
F_7	Roundness attached to a hook	å, ä, a, ö	0.12
F_8	Angularity open upward	v, y, x, u	0.12
F_9	Zigzaggedness	z, s, r, x	0.13
Features for Morse code data based on the 2-dimensional MDS solution by Shepard (1980)			
F_0	universal feature	All objects	1.11
F_1	1 component	E, T	1.25
F_2	2 components	A, I, M, N	0.90
F_3	3 components	D, G, K, O, R, S, U, W	0.40
F_4	4 components	B, C, F, H, J, L, P, Q, V, X, Y	0.00
F_5	5 components	1, 2, 3, 4, 5, 6, 7, 8, 9, 0	0.40
F_6	dots only	E, H, I, S, 5	0.49
F_7	1 dash, 1 dot	A, N	0.19
F_8	1 dash, 2 dots	D, R, U	0.10
F_9	1 dash, 3 dots	B, F, L, V	0.11
F_{10}	1 dash, 4 dots	4	0.15
F_{11}	2 dashes, 1 dot	G, K, W	0.00
F_{12}	2 dashes, 2 dots	C, P, Z	0.00
F_{13}	2 dashes, 3 dots	13, 3, 7	0.00
F_{14}	3 dashes, 1 dot	14, J, Q, Y	0.12
F_{15}	3 dashes, 2 dots	15, 2, 8	0.15
F_{16}	4 dashes, 2 dots	16, 1, 9	0.42
F_{17}	dashes only	0	0.63

2.5 Results simulation

Bias

Table 2.5 displays the bias and *rmse* for the ICLS and the OLS estimators, the bootstrap standard deviations of these estimates, and the nominal standard errors $\hat{\sigma}_{\text{ICLS}}$, $\hat{\sigma}_{\text{OLS}}$. The bias of the ICLS estimator, displayed in the first part of Table 2.5, is almost equal to the bias of the OLS estimator, except that the ICLS estimator has more bias for the parameters with values equal or close to zero, and for the intercept parameter. The variability of the ICLS estimator, expressed by the standard deviation, is in general equal to the variability of the OLS estimator, but lower for the (near) zero parameter

Table 2.5: Bias and *rmse* of $\hat{\eta}$, $\hat{\sigma}_{\hat{\eta}}$, and bootstrap standard deviation (sd_B) for OLS and ICLS estimators, resulting from the Monte Carlo simulation based on the *consonant* data.

	ICLS	OLS	ICLS	OLS	ICLS	OLS	ICLS	OLS	ICLS	OLS
η	mean $\hat{\eta}$		bias $\hat{\eta}$		rel. bias $\hat{\eta}$		sd $\hat{\eta}$		*rmse* $\hat{\eta}$	
2.23	2.19	2.23	-0.04	0.00	-0.02	0.00	0.11	0.13	0.12	0.13
1.21	1.21	1.20	0.00	-0.00	0.00	-0.00	0.11	0.11	0.11	0.11
0.78	0.78	0.78	-0.00	-0.00	-0.00	-0.00	0.13	0.13	0.13	0.13
0.37	0.37	0.37	0.01	0.01	0.02	0.02	0.11	0.11	0.11	0.11
0.09	0.11	0.09	0.01	0.00	0.14	-0.00	0.09	0.11	0.09	0.11
0.08	0.08	0.08	0.00	-0.01	0.00	-0.10	0.08	0.11	0.08	0.11
0.00	0.04	-0.00	0.04	-0.00	$-^*$	$-^*$	0.07	0.11	0.08	0.11
$\sigma_\eta^\ddagger$	mean $\hat{\sigma}_{\hat{\eta}}$		bias $\hat{\sigma}_{\hat{\eta}}$		rel. bias $\hat{\sigma}_{\hat{\eta}}$		sd $\hat{\sigma}_{\hat{\eta}}$		*rmse* $\hat{\sigma}_{\hat{\eta}}$	
0.14	0.14	0.13	0.00	0.00	0.01	-0.00	0.01	0.01	0.01	0.01
0.11	0.11	0.11	-0.00	0.00	-0.01	-0.00	0.01	0.01	0.01	0.01
0.13	0.13	0.13	-0.00	0.00	-0.00	-0.00	0.01	0.01	0.01	0.01
0.11	0.11	0.11	0.00	0.00	-0.00	-0.00	0.01	0.01	0.01	0.01
0.11	0.15	0.11	0.04	0.00	0.38	-0.00	0.06	0.01	0.08	0.01
0.11	0.15	0.11	0.04	0.00	0.32	-0.00	0.05	0.01	0.06	0.01
0.00	0.05	0.11	0.05	0.11	$-^*$	$-^*$	0.06	0.01	0.07	0.11
$\sigma_\eta^\ddagger$	mean sd_B		bias sd_B		rel. bias sd_B		sd sd_B		*rmse* sd_B	
0.14	0.12	0.13	-0.02	-0.00	-0.14	-0.01	0.01	0.01	0.02	0.01
0.11	0.11	0.11	-0.00	0.00	-0.01	-0.00	0.01	0.01	0.01	0.01
0.13	0.13	0.13	-0.00	-0.00	-0.01	-0.00	0.01	0.01	0.01	0.01
0.11	0.11	0.11	-0.00	0.00	-0.02	-0.00	0.01	0.01	0.01	0.01
0.11	0.08	0.11	-0.03	0.00	-0.23	-0.00	0.02	0.01	0.03	0.01
0.11	0.08	0.11	-0.03	-0.00	-0.30	-0.01	0.03	0.01	0.04	0.01
0.00	0.06	0.11	0.06	0.11	$-^*$	$-^*$	0.03	0.01	0.07	0.11

* The true values being equal to zero, the calculation of the relative bias leads to dividing by zero.

‡ σ_η stands for σ_{ICLS} and σ_{OLS} because in this particular case both true variability values are equal.

values. The *rmse* shows the same pattern: the *rmse* for both estimators are equal, with lower *rmse* of the ICLS for the (near) zero parameters. In sum, the ICLS estimator performs better than the OLS estimator in estimating the true value, which is to be expected in a situation where it is correct to apply constraints. The same conclusions for the ICLS and the OLS estimator can be drawn from the additional three simulation studies (the results are not shown in a table).

The second part of Table 2.5 provides information on the performance of the nominal standard errors, $\hat{\sigma}_{\text{ICLS}}$, $\hat{\sigma}_{\text{OLS}}$, compared to the true standard deviations σ_{ICLS} and σ_{OLS}. The estimator $\hat{\sigma}_{\text{OLS}}$ has no bias (except for the standard deviation associated with the parameter that has true value equal to zero), whereas $\hat{\sigma}_{\text{ICLS}}$ is clearly biased, especially for the (near) zero parameter values. The *rmse* of $\hat{\sigma}_{\text{ICLS}}$ is larger than the *rmse* of $\hat{\sigma}_{\text{OLS}}$ when the true values are almost equal or equal to zero. The results of the three other simulation studies show the same pattern: $\hat{\sigma}_{\text{ICLS}}$ is biased, while $\hat{\sigma}_{\text{OLS}}$ has no bias, and the *rmse* of $\hat{\sigma}_{\text{ICLS}}$ is larger than the *rmse* of $\hat{\sigma}_{\text{OLS}}$ when the true values are almost equal or equal to zero (the results are not shown in a table).

The last part of Table 2.5 shows the bootstrap standard deviations of the ICLS and the OLS estimators. In general, the empirical variability of the OLS estimator is almost equal to both the nominal variability and the true variability, which is to be expected from a consistent and unbiased estimator. The empirical variability of the ICLS estimator is smaller compared to both true and nominal variability. The bootstrap estimates of variability of this estimator have more bias and higher *rmse* values than the OLS estimator. Again, these conclusions hold without exceptions for the three remaining simulation studies (no results shown in a table).

Coverage

Table 2.6 displays the coverage proportions of the nominal and the empirical 95% confidence intervals for the ICLS and the OLS estimators, resulting from the simulation study based on the *consonant* data. The coverage of the OLS estimator is equal or very close to the nominal 95% level for all types of confidence intervals, nominal as well as empirical. The coverage of the ICLS estimator does not show the same consistent pattern as the OLS estimator: the nominal coverage is better than the empirical coverage , but it is sometimes too liberal with proportions exceeding the 95% level. There is no difference in performance between the bootstrap-*t* interval and the BC_a interval: both bootstrap intervals have inadequate coverage for the (near) zero parameter values. Apparently, the BC_a interval has some difficulties in correcting for the bias.

Compared to the simulation based on the consonant data, the three additional simulation studies have exactly the same results for the OLS estimator, but some differences appear for the ICLS estimator. The coverage proportions for the ICLS estimator obtained from the four simulation studies are summarized in Figure 2.6. (The plot showing the results of the *consonant* data is based on the proportions in Table 2.6, and the remaining plots are based on similar tables, which are not shown in this paper.) The simulation based on the *Morse code* data shows almost the same pattern as the simulation based on the *consonant* data. The most striking result is that the coverage performances of the BC_a intervals are poor when the parameter

values are equal or close to zero, but improve with higher parameter values. Both the nominal and the bootstrap-*t* confidence intervals perform better when parameter values are equal or close to zero. In the middle range of the parameter values, the three types of confidence intervals perform equally well with coverage proportions approaching the nominal 95% level. The same pattern of coverage results for all three confidence interval types, however in a lesser extent, can be seen in the plot of the simulation based on the *Swedish letters* data (Figure 2.6). The best results occur in the simulation based on the *similarity of faces* data: the three types of confidence intervals perform equally well by attaining the nominal 95% level for all parameter values. In this particular condition, there is only one true parameter value equal to zero, while all the other conditions have increasing numbers of true parameter values equal or close to zero.

Table 2.6: Coverage , empirical power and alpha for nominal and empirical 95% confidence intervals (Monte Carlo simulation based on *consonant* data)

	ICLS	OLS	ICLS	OLS	ICLS	OLS
Proportion coverage 95% confidence intervals						
η	Nominal CI		Bootstrap-*t* CI		BC_a CI	
2.226	0.98	0.95	0.94	0.95	0.93	0.95
1.206	0.96	0.95	0.95	0.95	0.95	0.95
0.778	0.95	0.95	0.95	0.95	0.95	0.95
0.366	0.95	0.95	0.94	0.95	0.95	0.95
0.092	0.97	0.95	0.89	0.95	0.82	0.95
0.084	0.99	0.96	0.86	0.95	0.98	0.95
0.000	0.98	0.95	0.97	0.94	0.92	0.94
Empirical power and alpha 95% confidence intervals						
	Empirical power					
η	Nominal CI		Bootstrap-*t* CI		BC_a CI	
2.226	1.00	1.00	1.00	1.00	1.00	1.00
1.206	1.00	1.00	1.00	1.00	1.00	1.00
0.778	1.00	1.00	1.00	1.00	1.00	1.00
0.366	0.92	0.92	0.93	0.92	0.98	0.92
0.092	0.14	0.14	0.15	0.14	0.43	0.14
0.084	0.07	0.10	0.09	0.10	0.11	0.10
	Empirical alpha					
η	Nominal CI		Bootstrap-*t* CI		BC_a CI	
0.00	0.02	0.05	0.03	0.06	0.08*	0.06

* $p < .05$; A 95 % CI around α, $[0.037, 0.064]$, can be obtained by considering each hypothesis test as a Bernoulli outcome, with $p = .05$, $q = 1 - p = .95$, $S = 1000$, and standard deviation $\sqrt{pq/S}$ (*cf.* Lee & Rodgers, 1998).

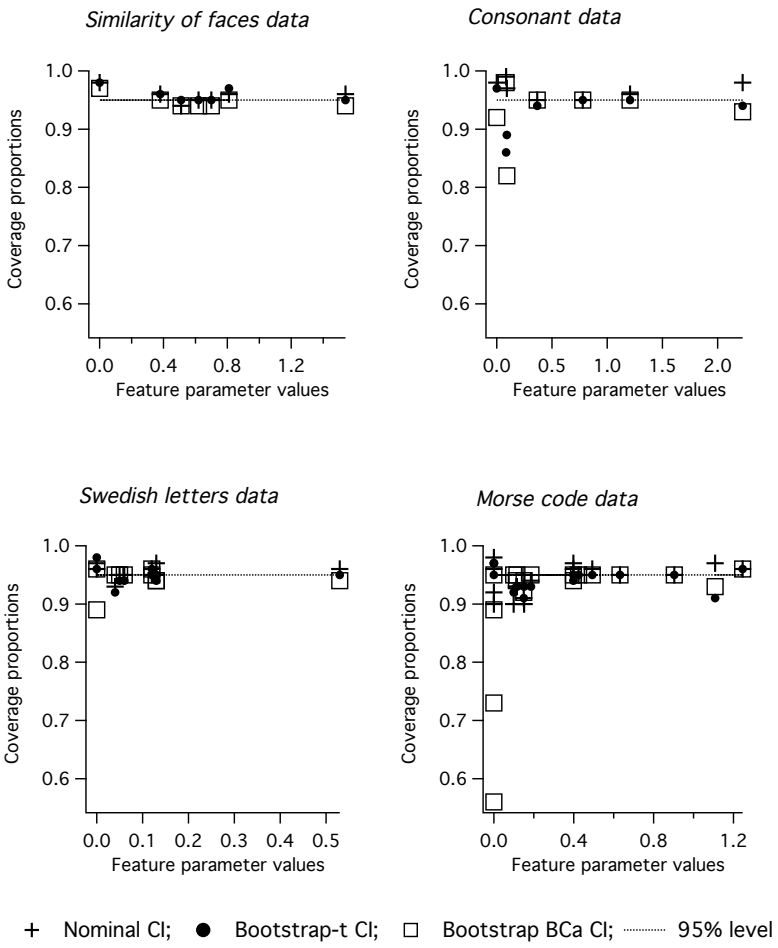

Figure 2.6: Coverage Nominal CI, Bootstrap-t CI, and BC_a CI for ICLS estimates for all simulation studies. The order of the plots follows the increasing number of zero and close to zero parameters present in the data.

The general conclusion seems to be that differences in performance of coverage between the three types of confidence intervals increase when there are more true parameter values equal or close to zero, and as a consequence, more constraints are activated. The most important differences arise in the simulation based on the *Morse code* data that has the largest number (= 4) of true parameter values equal to zero.

Power and alpha

Concerning the empirical power and the empirical alpha levels, as can be seen in Table 2.6, the empirical power is very high for the highest parameter values for both estimators; when the parameter values come closer to zero, power declines rapidly. This result does not only hold for the simulation based on the *consonant* data (as shown in Table 2.6), but also holds for the remaining three simulation studies.

The empirical alpha in the *consonant* data based simulation (Table 2.6) is very close to the nominal 5% level for the OLS estimator only, and holds for all confidence intervals. The empirical alpha levels for the ICLS estimator are too conservative when nominal confidence intervals and bootstrap-*t* confidence intervals are used. In contrast, the BC_a interval shows liberal empirical alpha levels. The same conclusions can be extended to the other simulation studies.

2.6 Discussion

In this paper we tried to construct a basis for statistical inference for the Feature Network Models by placing the models in the context of univariate (multiple) linear regression with positivity constraints on the parameters. We evaluated the performance of theoretical standard errors for the inequality constrained least squares estimator in comparison to its empirical variability.

In conclusion, the simulation studies show that the ICLS estimator is a better estimator than the OLS estimator, because it has has smaller *rmse* when true parameter values are positive. The nominal standard errors of the ICLS estimator are, however, larger than the empirical variability of this estimator. These larger values of the standard errors lead to liberal coverage proportions and to conservative empirical alpha levels. The nominal standard errors of the OLS estimator are very close to the empirical variability. The best coverage results, as well as the best results of empirical alpha, are achieved by the OLS estimator, and these results hold for all types of confidence intervals.

In case of the ICLS estimator, the worst coverage performances occur with the BC_a intervals, especially with increasing number of true parameter values equal or close to zero, and consequently more activated constraints. The bootstrap-*t* and the nominal confidence intervals perform better with increasing number of activated constraints: the results are equal (simulation based on *Swedish letters* data) or, sometimes, there are better results for the nominal confidence intervals (simulation based on *consonant* data), and sometimes the bootstrap-*t* intervals perform better (simulation based on *Morse code* data). However, the results of the bootstrap-*t* intervals are not that much better to entirely justify the computational costs.

The expectation is that when estimates are biased, the BC_a confidence intervals will perform better. The reason for the unsatisfactory results for BC_a intervals is not fully understood yet, and needs further investigation. A tentative explanation would be that, when constraints are activated very often, up to 50% of the sampling values of the ICLS estimates values are equal to zero. As a result, the sampling distribution is so much disturbed that the BC_a interval cannot adjust for it anymore, which results in confidence intervals that are too narrow.

The general conclusion is that one should be careful with the use of confidence intervals when several constraints are activated in the constrained least squares context. The results of a nonparametric bootstrap study in this situation can lead to the wrong conclusions about the coverage of the confidence intervals . The BC_a intervals are the least to be trusted. The nominal and the bootstrap-t intervals perform much better, with the nominal intervals having the advantage of no computational costs.

The confidence intervals in this study were used for coverage purposes and were not primarily intended for hypothesis testing. The same duality theory that serves as the basis for the estimation of the standard errors can also be applied to obtain a hypothesis test, where the null hypothesis of the inequality constraints (the constrained model) is tested versus an unrestricted alternative, using the Kuhn-Tucker test (Wolak, 1987). This test involves the calculation of weights that can be obtained in closed form for the cases were the number of predictor variables is less than 4. For more than 4 predictor variables, approximate weights can be obtained using Monte Carlo techniques. The requirement of this additional simulation step is the reason that we did not include the Kuhn-Tucker test in our simulation procedures.

There are several limitations in this study. Statistical inference was limited to the context of known features, which corresponds to a univariate (multiple) regression problem with a fixed set of predictor variables. The case of unknown features necessitates a different framework for statistical inference because the predictor variables become random variables. The simulation study is limited to the situation where the assumptions for statistical inference in linear regression hold. Additional simulation studies are needed to evaluate the performance of the theoretical standard errors under violation of the assumptions (e.g. skewed distributed variables).

Similar results can be obtained for standard errors in additive trees (Frank & Heiser, 2004) and are expected to hold for ultrametric trees also. In either case, the distances follow the path-length metric, which, when defined on the tree structure, may be viewed as an additive version of the distinctive features model of dissimilarity. Furthermore, Carroll and Corter (1995) have shown that clusterings with associated weights estimated using the common features model can be represented by ultrametric, additive and extended trees or multiple trees, when the distances are defined as path lengths between objects. So in principle, a simple adjustment of the FNM yields the standard errors for all these models. It should be noted, that the reverse process, the representation of distinctive features models by common features models, still faces problems of non-uniqueness. However, for the ADCLUS model the current theory can be applied directly on the cross product terms of the feature indicators for all object pairs, and therefore, our results are easily extended to this model as well.

Even in models not primarily related to the FNM, but having the same inequality

constrained least squares context, like for example the latent budget model (Mooijaart, van der Heijden, & van der Ark, 1999) or the Q-matrices in the rule space model for cognitive diagnosis (Tatsuoka, 1995), the results of this study are expected to hold.

We noticed that the constraints are not activated very often, which means that, most of the time the ICLS estimates reduce to the OLS estimates. Therefore, we believe that statistical inference for the Feature Network Models can benefit from the nice statistical properties of the OLS estimator.

Chapter 3

Standard Errors, Prediction Error and Model Tests in Additive Trees [1]

Abstract

Theoretical standard errors and confidence intervals are given for the estimates of branch lengths in psychometric additive trees for a priori known tree topologies as well as for estimated tree topologies. A model test and an estimate of prediction error to compare different tree topologies are also given. The statistical inference theory proposed here differs from existing approaches due to the combination of the use of features with the multiple regression framework. Additive trees can be considered as a special case of Feature Network Models, where the objects are described by features, which are binary variables that indicate whether a particular characteristic is present or absent. Considering features as predictor variables leads in a natural way to the univariate multiple regression model.

3.1 Introduction

In general, there are two types of graphical representations of proximity data: spatial models and network models. The spatial models - such as multidimensional scaling - represent each object as a point in a coordinate space (usually Euclidean space) in such a way that the metric distances between the points approximate the observed proximities between the objects as closely as possible. In network models, the objects are represented as nodes in a connected graph, so that the spatial distances between the nodes in the graph approximate the observed proximities among the objects. In MDS, the primary objective is to find optimal coordinate values that lead to distances that approximate the observed proximities between the objects, whereas in network models, the primary objective is to find the correct set of *relations* between the objects that describe the observed proximities.

[1] This chapter has been submitted for publication as: Frank, L. E. & Heiser, W. J. (2005). Standard errors, prediction error and model tests in additive trees. *Submitted manuscript.* With an exception for the notes in this chapter, which are reactions to remarks made by the members of the promotion committee.

Feature Network Models or FNM (Heiser, 1998) represent proximity data in a discrete space, usually by a network representation. The relations between the objects are characterized by the kind of features they possess and by the combination of these features. Features are binary variables indicating for each object whether a particular characteristic is present or absent. The relations between the features, or the feature structure, determine the shape of the graphical representation, which is either a network or a tree. Therefore, FNM can be viewed as a general framework for graph representations, where the network is the general case and trees are special cases.

In FNM, the relation between any two objects i and j is represented by the symmetric set difference (= the difference between the union and the intersection of two sets) of the set of features that describes the two objects. The symmetric set difference expresses the number of features that object i possesses that are not shared by object j and vice versa, which amounts to the number of non-common elements of the objects. Applying the symmetric set difference on binary features in a binary coordinate space, corresponds to the *Hamming* distance, or the city-block distance. The relation between the objects can be expressed in terms of city-block distances, which is useful for graphical display purposes. Besides the graphical representation, the features in their own right are highly informative about the relations between the objects. In the final solution, each feature has a parameter value that indicates its relative importance: the *feature discriminability* value.

Since the introduction by Tversky (1977) of the Contrast Model, where objects are represented by subsets of discrete features, several different tree models have been developed in the psychological literature that are based on features (see Carroll & Corter, 1995, and Corter, 1996 for an overview). These models neither provide ways to estimate the standard errors of the parameter values, nor provide confidence intervals to assess the stability of the solution. In some psychometric applications (e.g. De Soete, 1983; Corter, 1996) least squares minimization is used to obtain the solution, treating the problem as a multiple regression model. Nevertheless, in the psychological literature, the statistical inference aspects of the multiple regression model have not been fully exploited for additive trees. The statistical inference theory proposed in this paper derives from the multiple regression framework because the use of features, when considered as predictor variables, leads in a natural way to the univariate multiple regression model. However, the standard multiple regression statistical inference theory cannot be applied because the network or additive tree representation imposes constraints on the model parameters. Negative edge lengths have no meaning in a network or an additive tree. In the context of FNM the implication is that the feature discriminability parameters associated with the features (the predictor variables) are constrained to be positive. These positivity constraints are even more relevant for additive tree representations because each branch in the tree is represented by a single feature, as will become clearer in this paper.

In contrast to the psychological tree domain, the phylogenetic tree domain does have a strong tradition of statistical inference. Important contributions in the field of statistical inference in phylogenies were made by Felsenstein (1985 and, for an overview, 2004, Chapters 19 - 21) and by Nei, Stephens, and Saitou (1985). Felsenstein (1983) evaluated the stability of a tree topology using the bootstrap to calculate

the proportion of bootstrap trees that agree with the original tree in terms of topology and not directly in terms of branch lengths. In addition, the phylogenetic literature offers many examples of the estimation of the standard errors of branch lengths. The branch lengths are usually estimated with ordinary least squares, and the variances of the branch lengths are calculated by taking into account the method used to compute the evolutionary distances (Li, 1989; Nei et al., 1985; Rzhetsky & Nei, 1992; Tajima, 1992). Bulmer (1991) estimated the branch lengths and their standard errors with generalized least squares, which allows for correcting the correlation of distances between species that share one or more common paths. Despite the abundance of methods to compute standard errors for the branches of the phylogenetic trees, none of these methods take into account that when estimating the standard errors of the branch length estimates, one should correct for the fact that the estimates of the branch lengths have been constrained to be positive. The problem of biased estimates of the branch lengths has been diagnosed by Gascuel and Levy (1996), who correctly remark that the right way to estimate the edge lengths in phylogenies is to use linear regression under positivity constraints, and by Ota, Waddell, Hasegawa, Shimodaira, and Kishino (2000), who use a mixture of χ^2 distributions to construct appropriate likelihood ratio tests for nested evolutionary tree models. The mixture of χ^2 distributions is based on earlier results obtained by Self and Liang (1987) and Stram and Lee (1994) who derived limiting distributions of the likelihood ratio statistic when varying numbers of parameters are on the boundary. However, in the additive tree framework, Ota et al. (2000) have not made adjustments for the estimation of the standard errors of the branch lengths.

Recently, Frank & Heiser (in press *a*) showed how to compute standard errors and confidence intervals for the inequality constrained feature discriminability parameters in FNM. In this paper, we will show that the same statistical inference theory that has been proven to be useful for networks also applies to the family of tree representations. We propose a way to compute standard errors and confidence intervals for branch lengths of additive trees, and especially for tree topologies that include star shaped components, which means that one or more branches have edge lengths equal to zero (resulting from the correction of negative values). The multiple regression framework can be used to impose inequality constraints on the parameters and at the same time to compute theoretical standard errors for the inequality constrained least squares parameters that represent the edge lengths of the branches in an additive tree. These standard errors were introduced by Liew (1976) and take into account the fact that the parameter estimates are bounded below by zero. Whereas the results presented by Frank & Heiser (in press *a*) were limited to the situation of an a priori known feature structure (or tree topology), the present study shows that the same theory can be applied for the situation where the tree topology is not known in advance if the sample can be divided in a test set and a training set. Resulting from the same inequality constrained least squares framework, the paper shows an application of the Kuhn-Tucker test that is used to test whether the constrained solution is in accordance with the data. In addition, an easy way to estimate the prediction error of the model is provided, which allows for comparison of different tree topologies.

The remainder of this paper is organized as follows. It starts with a description of

the Feature Network Models with an application on sample data, followed by an explanation of additive trees as special cases of FNM. Next, the statistical inference theory for inequality constrained least squares is introduced and evaluated with Monte Carlo simulation techniques. The first simulation study shows how to obtain the empirical *p*-value for the Kuhn-Tucker test. The second simulation study assesses the performance of the theoretical standard errors in comparison to bootstrap standard errors for the case where the tree topology is known in advance. It will become clear that the theoretical standard errors are much closer to the true values than the bootstrap standard errors and that the confidence intervals based on theoretical standard errors have better coverage performance than the bootstrap confidence intervals. The third simulation study shows that the same statistical inference theory can be applied in the situations where the tree topology is not known in advance and estimated with the neighbor-joining (NJ) method (Saitou & Nei, 1987). The NJ method is a widely used tree finding algorithm, especially in the phylogenetic domain, that is related to the ADDTREE algorithm by Sattath and Tversky (1977), which was developed in the mathematical psychology domain. Saitou and Nei (1987) and Gascuel (1994) have demonstrated that the NJ and the ADDTREE algorithms are strongly related and usually provide identical or very similar trees. A comparison between the statistical inference theory proposed for FNM in this paper and the statistical inference practice in the phylogenetic tree domain is provided in the discussion.

3.2 Feature Network Models

Since the general framework of this paper is the network representation, this section starts with a description of the Feature Network Models. FNM represent proximity data in a discrete space usually by a network representation. The properties of the models will be illustrated using a data set, the *kinship* data of Rosenberg and Kim (1975). A number of 165 female students and 165 male students were asked to group fifteen kinship terms on the basis of their similarities in minimally two and maximally fifteen categories. Half of the students were allowed to do the sorting task more than one time. Dissimilarity measures were derived for each pair of kinship terms by counting the number of subjects who placed the two terms in different categories. The data that were used in this study are the dissimilarity values of the female students ($n = 165$). Analyzing the dissimilarity matrix for the female students with the cluster differences scaling algorithm[2] of FNM (Heiser, 1998) yielded a solution with 5 features, displayed in Table 3.1. The features represent criteria most likely used by the female students to categorize the kinship terms.

Features are binary variables indicating for each object whether a particular characteristic is present or absent. Some set theoretic properties of the binary feature matrix lead to the estimation of a distance measure that approximates the observed dissimilarities. The difference between the union and intersection (= the symmetric set

[2]The first application of FNM used a cluster differences scaling algorithm (Heiser, 1998) with number of clusters equal to two, which constitutes a one-dimensional MDS problem with the coordinates restricted to form a bipartition. Because it is still a hard combinatorial problem, the implementation uses a nesting of several random starts together with *K*-means type of reallocations.

Table 3.1: The 5 binary features describing the kinship terms

Kinship terms	Gender	Nuclear family	Collaterals	Generation(1, 2)	Parent/child
aunt	0	0	1	1	0
brother	1	1	1	0	0
cousin	1	0	0	0	0
daughter	0	1	1	0	1
father	1	1	1	1	1
granddaughter	0	1	0	1	1
grandfather	1	1	0	1	0
grandmother	0	1	0	1	0
grandson	1	1	0	1	1
mother	0	1	1	1	1
nephew	1	0	0	0	1
niece	0	0	0	0	1
sister	0	1	1	0	0
son	1	1	1	0	1
uncle	1	0	1	1	0

difference) expresses the number of non-common features possessed by the objects i and j. For example, the symmetric set difference for the two kinship terms *aunt* and *cousin* is the set $\{Gender, Collaterals, Generation\}$. Following Goodman (1951, 1977) and Restle (1959, 1961), a distance measure that satisfies the metric axioms can be expressed as a simple count μ of the elements of the symmetric set difference between the stimuli O_i and O_j and becomes the *feature distance*: $d(O_i, O_j) = \mu[(O_i \cup O_j) - (O_i \cap O_j)]$.

If $\mathbf{E}$ is a binary matrix of order $m \times T$ that indicates which features t describe the m objects, as in Table 3.1, the re-expression of the feature distance in terms of coordinates is as follows (Heiser, 1998):

$$\begin{aligned} d(O_i, O_j) &= \mu[(O_i \cup O_j) - (O_i \cap O_j)] \\ &= \sum_t |e_{it} - e_{jt}|, \end{aligned} \tag{3.1}$$

where $e_{it} = 1$ if feature t applies to object i, and $e_{it} = 0$ otherwise. This re-expression of the feature distance in terms of binary coordinates is also known as the *Hamming* distance. The feature distance used in FNM is a weighted version of the distance in Equation 3.1:

$$d(O_i, O_j) = \sum_t \eta_t |e_{it} - e_{jt}|, \tag{3.2}$$

where the weights η_t express the relative contribution of each feature. Each feature splits the objects into two classes, and η_t measures how far these classes are apart.

Table 3.2: Feature parameters ($\hat{\eta}$), standard errors and 95% t-confidence intervals for Feature Network Model on *kinship* data with $R^2 = .95$.

Features	$\hat{\eta}$	$\hat{\sigma}_{\eta}$	95% CI	
Gender	27.54	0.63	26.31	28.77
Nuclear family	25.22	0.66	23.93	26.51
Collaterals	21.71	0.64	20.46	22.96
Generation (1,2)	18.58	0.64	17.33	19.83
Parent/child	15.06	0.64	13.81	16.31

For this reason, Heiser (1998) called the feature weight a *discriminability parameter*. The feature discriminability parameters are estimated by minimizing the following least squares loss function:

$$\min_{\hat{\boldsymbol{\eta}}} = \|\mathbf{X}\hat{\boldsymbol{\eta}} - \boldsymbol{\delta}\|^2, \tag{3.3}$$

where $\mathbf{X}$ is of size $n \times T$ and $\boldsymbol{\delta}$ is a $n \times 1$ vector of dissimilarities, with n equal to all possible pairs of m objects: $m(m-1)/2$. The problem in Equation 3.3 is expressed in a more convenient multiple linear regression problem, where the matrix $\mathbf{X}$ is obtained by applying the following transformation on the rows of matrix $\mathbf{E}$ for each pair of objects, where the elements of $\mathbf{X}$ are defined by:

$$e_{lt} = |e_{it} - e_{jt}|, \tag{3.4}$$

where the index $l = 1, \cdots, n$ varies over all pairs (i, j). The result is the binary $(0, 1)$ matrix $\mathbf{X}$, where each row ($\mathbf{x}'$) represents the distinctive features for some pair of objects, with 1 meaning that the feature is distinctive for a pair of objects. The weighted sum of these distinctive features is the fitted distance for each pair of objects and is equal to $\mathbf{d} = \mathbf{X}\boldsymbol{\eta}$. Corter (1996, Appendix C, p. 57) uses a similar matrix $\mathbf{X}$ in the linear regression context to obtain the lengths of the branches in an additive tree.

Table 3.2 shows the feature discriminability parameters $\hat{\eta}_t$ obtained by PROXGRAPH, the program developed in Matlab to fit the FNM. The five features solution explains 95.35% of the variance in the data, and the values of the feature parameters lead to the conclusion that the most important categorizing criteria were: *Gender*, *Nuclear family*, and *Collaterals*. All five features played a more or less important role in categorizing the kinship terms as follows from the 95% t-confidence intervals that show that all feature parameters differ significantly from zero (Table 3.2).

Figure 3.1 shows the Feature Network representation that results from the fitted distances on the *kinship* data. The kinship terms are the vertices in the network and the feature distances ($\hat{\mathbf{d}} = \mathbf{X}\hat{\boldsymbol{\eta}}$) are represented as the sum of the edge lengths along the shortest path in the graph, where the edge lengths are the feature parameters $\hat{\boldsymbol{\eta}}$. How the network is obtained will be explained in the following section. The five-dimensional feature network has been embedded in 3-dimensional Euclidean space using PROXSCAL[3], a multidimensional scaling program distributed as part of

[3] with the interval transformation option and initialized with the simplex solution

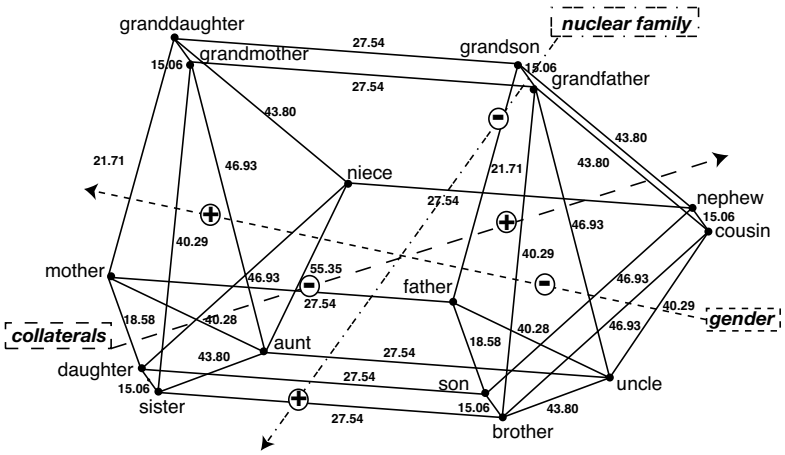

Figure 3.1: Feature Network representation for the *kinship* data with the three most important features (*Gender*, *Nuclear family* and, *Collaterals*) represented as vectors. The plus and minus signs designate the projection onto the vector of the centroids of the objects that posses the feature (+) and the objects that do not have that feature (-).

the Categories package by SPSS (Meulman & Heiser, 1999). The solution of the common space was restricted by a linear combination of the feature variables that are represented as vectors in Figure 3.1, leading from the origin through the point with coordinates equal to the correlations of each feature with each of the three dimensions. The network clearly shows the distinction between the female kinship terms and the male kinship terms produced by the most important feature *Gender*. This feature as well as the second and third most important features *Nuclear family* and *Collaterals* are represented by vectors in the network. The plus and minus signs on each vector designate the projection onto the vector of the centroids of the kinship terms that posses the feature $(+)$ and the kinship terms that do not possess that feature $(-)$.

3.3 Feature Network Models: network and additive tree representations

The relations between the features in FNM determine the shape of the network. A set of overlapping features will result in a network graph, which is a connected graph with cycles. When the set of features has a nested structure, i.e., all pairs of features are either nested or disjoint, the network will have the shape of an unrooted additive tree, a graph without cycles (Buneman, 1971). If the unrooted additive tree is a bifurcating tree, there are fixed numbers of edges (branches), internal and external nodes, given a number of objects m (*cf.* Felsenstein, 2004, Chapter 3). Bifurcating trees have interior nodes of degree 3, meaning that each internal node connects to

	Features								
	F1	F2	F3	F4	F5	F6	F7	F8	F9
Objects									
a	1	1	0	1	0	0	0	0	0
b	1	1	0	0	1	0	0	0	0
c	1	0	1	0	0	1	0	0	0
d	1	0	1	0	0	0	1	0	0
e	0	0	0	0	0	0	0	1	0
f	0	0	0	0	0	0	0	0	1
Internal nodes									
(e,f)	0	0	0	0	0	0	0	0	0
(a,b,c,d)	1	0	0	0	0	0	0	0	0
(a,b)	1	1	0	0	0	0	0	0	0
(c,d)	1	0	1	0	0	0	0	0	0

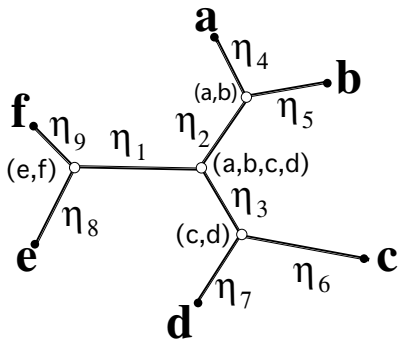

Figure 3.2: Nested and disjoint feature structure and corresponding additive tree representation. Each edge in the tree is represented by a feature and the associated feature discriminability parameter η_t.

three other nodes (internal or external) and every external node (or leaf node) is of degree 1, which means that only one branch leads to an external node. Given these specifications, the bifurcating unrooted additive tree for m objects has a number of $2m-3$ edges because for each new object added to an existing tree an internal node and two new edges must be added (*cf.* Felsenstein, 2004, Chapter 3). Following this reasoning, the number of internal nodes is fixed to $(2m-3-1)/2$. In contrast to the bifurcating trees, the multifurcating trees do not have a fixed number of edges and nodes for a given number of objects. Since the degree of each internal node in multifurcating trees is not necessarily equal to 3, there exists a range of possible numbers of internal nodes and numbers of edges that depend on the number of internal nodes.

In terms of features, the bifurcating unrooted additive tree has a set of $T = 2m-3$ nested features and the internal nodes are represented by $(T-1)/2$ supplementary objects added to the original set of objects in the feature matrix. Figure 3.2 shows the feature matrix and the corresponding tree graph for an example of 6 objects. There are $T = 2m-3 = 9$ nested features, $m = 6$ leaf nodes and, $n_o = (T-1)/2 = 4$ internal nodes. The nested structure of the features becomes apparent: the features either exclude each other or one is a subset of the other. Each cluster in the tree, for example the bipartition of the objects *a* and *b* against the other objects, is represented by a *cluster feature*, that is, a feature which describes more than one object, in this example feature F_2. The internal nodes are defined as supplementary objects with a feature pattern that is the intersection of the feature patterns of a subset of the objects. Therefore, they can be labeled by listing the objects in the subset. The leaf nodes of the tree represent the 6 objects and the associated edges correspond to *unique features*, which are features that belong to one object exclusively. In the example in Figure 3.2

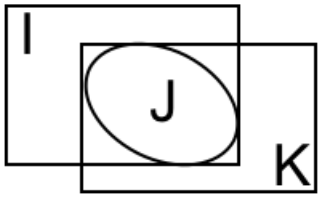

Figure 3.3: Betweenness holds when $J = I \cap K$, where I, J, and K are sets of features describing the corresponding objects i, j, and k.

the unique features are the set $\{F_4, F_5, F_6, F_7, F_8, F_9\}$. Note that these unique features are also either nested or disjoint with respect to the cluster features.

Additive tree representation and feature distance

The feature distance parallels the path-length distance in a valued graph when one of the metric axioms, the triangle inequality, is reaching its limiting additive form $d_{ik} = d_{ij} + d_{jk}$ (Flament, 1963; Goodman, 1951, 1977; Heiser, 1998). In a network graph, each time that the distance d_{ik} is exactly equal to the sum $d_{ij} + d_{jk}$ the edge between the objects i and k can be excluded, resulting in a parsimonious subgraph of the complete graph.

In terms of features the condition $d_{ik} = d_{ij} + d_{jk}$ is reached when object j is *between* objects i and k. The objects can be viewed as sets of features: $\mathcal{S}_i$, $\mathcal{S}_j$, and $\mathcal{S}_k$. Betweenness of $\mathcal{S}_j$ depends on the following conditions (Restle, 1959):

1. $\mathcal{S}_i$ and $\mathcal{S}_k$ have no common members which are not also in $\mathcal{S}_j$;
2. $\mathcal{S}_j$ has no unique members which are in neither $\mathcal{S}_i$ nor $\mathcal{S}_k$.

Figure 3.3 clearly shows that betweenness holds when the set $\mathcal{S}_j$ is exactly equal to the intersection of the sets $\mathcal{S}_i$ and $\mathcal{S}_k$ - in that case $\mathcal{S}_j$ has no unique features (Tversky & Gati, 1982) -, or when the set $\mathcal{S}_j$ consists of a subset of the intersection of the sets $\mathcal{S}_i$ and $\mathcal{S}_k$. In both situations $d_{ik} = d_{ij} + d_{jk}$. In the following, it will become clear that an additive tree structure results from a special feature structure where there always is an internal node $\mathcal{S}_j$ between any two leaf nodes $\mathcal{S}_i$ and $\mathcal{S}_k$.

An additive tree is a special subgraph of the complete graph, where each edge is represented by a separate feature. The edges leading directly to leaf nodes correspond to unique features, the set of features that describe only one object (see Figure 3.2). A nested set of features is not sufficient to produce a tree graph with FNM. A set of internal nodes has to be added to the set of objects (the external nodes). These internal nodes play the role of the set $\mathcal{S}_j$ in the betweenness condition by forcing the betweenness to hold exactly for any pair of objects i and k that have an associated nested set of features, leaving only paths between objects that are in an hierarchical relation to each other. Each edge between two internal nodes corresponds exactly to one cluster feature, and the edge length to its weight (see Figure 3.2).

It should be noted that the estimated distances between the internal nodes in the tree cannot be compared to dissimilarities because these quantities are not observed. To calculate all distances simultaneously requires a modification of the original feature matrix $\mathbf{E}$ (Equation 3.1). The feature matrix $\mathbf{E}$ is augmented with a supplementary set of objects equal to the number of internal nodes.

The augmented $\mathbf{E}_T$ matrix is as follows:

$$\mathbf{E}_T = \begin{bmatrix} \mathbf{E}_C & \mathbf{E}_U \\ \mathbf{E}_N & \mathbf{E}_0 \end{bmatrix}, \tag{3.5}$$

where $\mathbf{E}_C$ is a $m \times T_C$ matrix, representing the set of cluster features and $\mathbf{E}_U$ is a $m \times T_U$ matrix representing the set of unique features. Both parts describe the set of observed objects. The remaining two parts are related to the set of internal nodes (n_o): $\mathbf{E}_N$ is of size $n_o \times T_C$ and $\mathbf{E}_0$ contains zeros only and has size $n_o \times T_U$. Each row of $\mathbf{E}_N$ and $\mathbf{E}_0$ represents the feature pattern of each node. This nodal feature pattern is equal to the intersection of the feature patterns belonging to the objects (the rows of $\mathbf{E}_C$ and $\mathbf{E}_U$) that are represented by each particular node. The intersection of the feature patterns related to the unique features is always zero and, consequently, $\mathbf{E}_0$ contains zeros only. Figure 3.2 shows the four parts of the augmented $\mathbf{E}_T$ matrix. The objects (a, b, c, d, e, f) are described with cluster features and with unique features: the part with the cluster features, $\mathbf{E}_C$, is formed by the set features $\{F_1, F_2, F_3\}$, the part of the unique features, $\mathbf{E}_U$, is formed by $\{F_4, F_5, F_6, F_7, F_8, F_9\}$. The feature patterns of the internal nodes are represented by the parts $\mathbf{E}_N$ and $\mathbf{E}_0$. The $\mathbf{E}_0$ part is related to the unique features and contains zero's only. The $\mathbf{E}_N$ relates to the cluster features and the feature pattern of each internal node is formed by taking the intersection of the feature pattern belonging to the corresponding objects. For example, the feature pattern for internal node (a, b) is formed by taking the intersection of the feature pattern for object $a = \{110\}$ and object $b = \{110\}$, resulting in the feature pattern $\{110\}$.

Dissimilarities are only available for the objects and not for the internal nodes. Therefore, the feature discriminability parameters $\boldsymbol{\eta}$ are estimated using only the parts $\mathbf{E}_C$ and $\mathbf{E}_U$. After applying the featurewise distance transformation in Equation 3.4 to the matrix $[\mathbf{E}_C \quad \mathbf{E}_U]$, the resulting matrix $\mathbf{X}$ is used to obtain the estimates of the feature discriminability parameters ($\hat{\boldsymbol{\eta}}$) by minimizing the loss function in Equation 3.3. To obtain the estimated distances for the edges that are linked to internal nodes, the featurewise distance transformation (Equation 3.4) is applied to the augmented matrix $\mathbf{E}_T$, yielding the matrix $\mathbf{X}_T$. The estimated feature distances for the complete tree are equal to $\hat{\mathbf{d}}_T = \mathbf{X}_T\hat{\boldsymbol{\eta}}$. Given this description, it is easy to understand that every tree topology, known by theory or resulting from any tree constructing algorithm, can be transformed into an augmented feature matrix $\mathbf{E}_T$, such that, when analyzed as FNM with PROXGRAPH, it will lead to a tree representation of the data.

Example of additive tree obtained with feature structure

An example of a multifurcating additive tree is the solution obtained by De Soete and Carroll (1996) on the *kinship* data. The augmented $\mathbf{E}_T$ based on this given tree

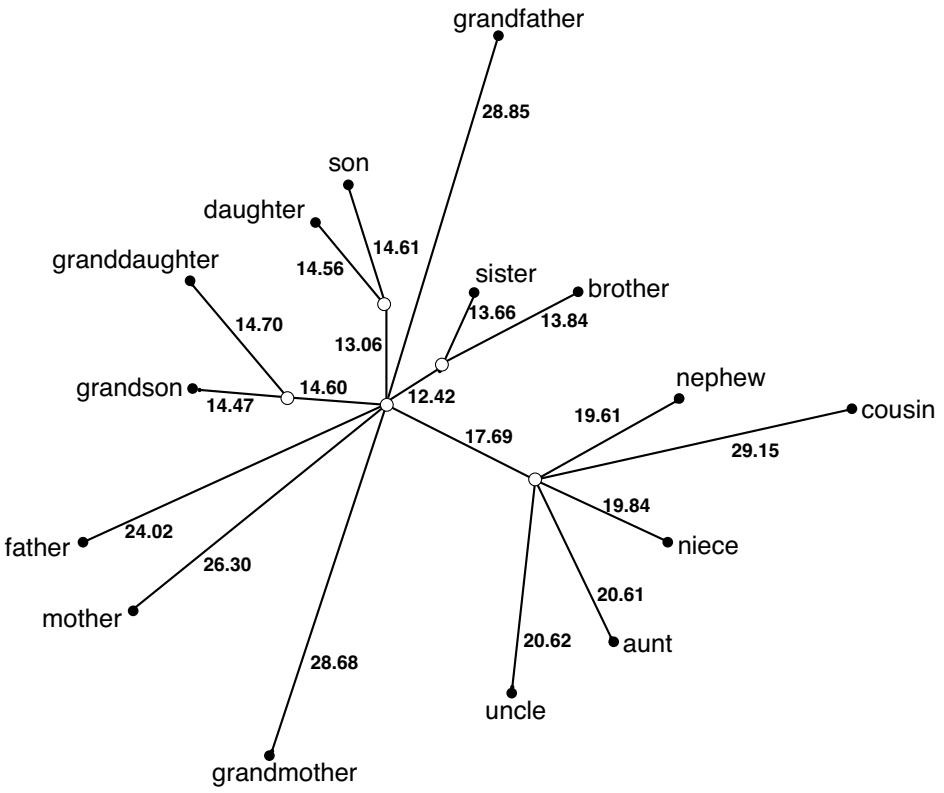

Figure 3.4: Unresolved additive tree representation of the *kinship* data based on the solution obtained by De Soete & Carroll (1996).

topology is displayed in the first part of Figure 3.5 and yields the additive tree representation in Figure 3.4. The 2-dimensional embedding of the tree has been obtained by submitting Euclidean distances calculated on the augmented $\mathbf{E}_T$ to the MDS program PROXSCAL[4], a multidimensional scaling program distributed as part of the Categories package by SPSS (Meulman & Heiser, 1999). The associated feature parameters and 95% *t*-confidence intervals are given in Figure 3.6. The construction of the confidence intervals will be explained in the next section. Some of the feature parameters have zero values ($F_2, F_3, F_4, F_{21}, F_{23}$) leading to the unresolved tree representation of Figure 3.4. The expected number of nodes is $12 + 1 = 13$ with 15 objects, but only 6 internal nodes remain in the final solution due to activation of the positivity constraints. The feature structure ($\mathbf{E}_T$) can therefore be simplified to the matrix shown in the second part of Figure 3.5.

[4]allowing a ratio scale transformation with a simplex start.

Feature structure resolved tree

	Nuclear family								Collaterals											Gandparents/Grandchildren							
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27
aunt	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
brother	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cousin	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
daughter	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
father	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
granddaughter	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
grandfather	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1
grandmother	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
grandson	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
mother	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nephew	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
niece	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
sister	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
son	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
uncle	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
nodes																											
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Feature structure unresolved tree

	Nuclear family + Grandparents/Grandchildren															Collaterals									
	F1	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F22	F24	F25	F26	F27			
aunt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0			
brother	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0			
cousin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0			
daughter	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
father	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0			
granddaughter	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
grandfather	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0			
grandmother	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0			
grandson	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
mother	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0			
nephew	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1			
niece	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0			
sister	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0			
son	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
uncle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0			
nodes																									
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
10	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
12	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
13	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0			

Figure 3.5: Feature structure for the resolved additive tree representation (*top*) of the *kinship* data and simplified feature structure for the unresolved additive tree representation (*bottom*) of Figure 3.4.

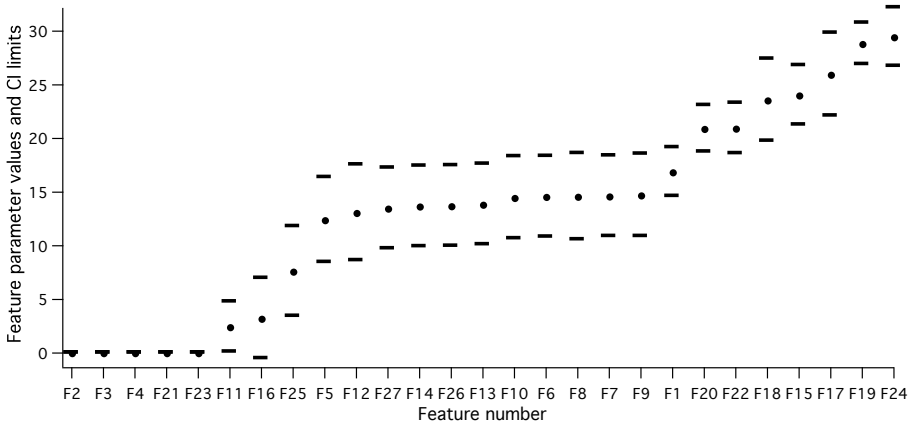

Figure 3.6: Feature parameters ($\hat{\eta}_{\text{ICLS}}$) and 95% t-confidence intervals for additive tree solution on *kinship* data with $R^2 = .96$.

3.4 Statistical inference in additive trees

This section shows how the multiple linear regression framework can be used to obtain several statistical inference measures for additive trees. The features of an additive tree can be considered as predictor variables and the feature discriminability parameters are estimated like regression coefficients, with the major difference that positivity constraints are imposed on the feature discriminability parameters, because they represent edge lengths in the tree representation. This section shows how to obtain standard errors for the inequality constrained least squares estimators that can be used to construct 95% t-confidence intervals for the feature discriminability parameters. The statistical inference theory is intended for the case where the tree topology is known in advance, but can also be applied when the tree topology is unknown, as will be shown in the following. This section also provides an application of the Kuhn-Tucker test that is used to test whether the constrained solution is in accordance with the data and results from the same theory used to obtain the standard errors. The last topic of this section provides a way to estimate prediction error with the *generalized cross-validation (GCV)* statistic. This estimate of prediction error combines the the analytical approximation of leave-one-out cross-validation commonly used in linear fitting methods with the inequality constrained least squares theory.

Obtaining standard errors for additive trees

An important difference of the current approach compared to what is usually done in the phylogenetic domain is that phylogenetic trees do not use explanatory variables like the features. In the case that the feature structure is known, the distinctive-feature additivity allows for considering the additive tree as a univariate multiple

linear regression model:

$$\boldsymbol{\delta} = \mathbf{X}\boldsymbol{\eta} + \boldsymbol{\epsilon} \tag{3.6}$$

where $\boldsymbol{\delta}$ is a $n \times 1$ vector with dissimilarities, $\mathbf{X}$ is a known $n \times T$ binary $(0, 1)$ matrix of rank T and $\boldsymbol{\eta}$ is a $T \times 1$ vector. Each row of the matrix $\mathbf{X}$ results from the operation $\mathbf{x}_l = |\mathbf{e}_{it} - \mathbf{e}_{jt}|$ (Equation 3.4). For an additive tree representation, $\mathbf{X}$ contains the featurewise distances that result from the matrix $\mathbf{E}_T$ formed by the set of cluster features and unique features, as explained in the previous section.

We assume, like Ramsay (1982), that $\boldsymbol{\epsilon}$ in Equation 3.6 is a $n \times 1$ random vector that follows a normal distribution with constant variance σ^2 over replications of judgments,

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mu, \sigma^2\mathbf{I}), \tag{3.7}$$

where $\mathbf{I}$ is an identity matrix of rank n, and where it is assumed that σ^2 is small enough to ensure the occurrence of negative dissimilarities to be negligible. The parameters of the vector $\boldsymbol{\eta}$ are subject to positivity constraints because they represent edge lengths of the tree. As explained in the beginning of this section, the phylogenetic domain does not apply positivity constraints when estimating branch lengths and trees that yield negative branch length estimates are simply discarded. Hence, the phylogenetic domain might benefit from the following theory on inequality constrained least squares estimation.

The inequality constrained least squares estimator $\hat{\eta}_{\text{ICLS}}$ results from the quadratic programming problem (*cf.* Björk, 1996):

$$\begin{aligned} \min_{\boldsymbol{\eta}} &= (\boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta})'(\boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta}) \\ &\text{subject to } \mathbf{A}\boldsymbol{\eta} \geq \mathbf{r}, \end{aligned} \tag{3.8}$$

where the matrix of constraints $\mathbf{A}$ is a $C \times T$ matrix of rank C, and $\mathbf{r}$ is a $C \times 1$ null-vector because all parameters are constrained to be greater than or equal to zero.

The duality theory of the quadratic programming problem of Equation 3.8 is the basis for the estimation of the standard errors of the parameters (Liew, 1976) and results in the following expression of the estimator $\hat{\eta}_{\text{ICLS}}$ in terms of the dual solution:

$$\hat{\boldsymbol{\eta}}_{\text{ICLS}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\delta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{A}'\tfrac{1}{2}\boldsymbol{\lambda}_{\text{KT}}, \tag{3.9}$$

where $\boldsymbol{\lambda}_{\text{KT}}$ is the vector with Kuhn-Tucker multipliers that results from solving the quadratic programming problem with Algorithm AS 225 (Wollan & Dykstra, 1987). As shown by Liew (1976) the estimated standard errors for the ICLS estimator vector are

$$\hat{\sigma}_{\text{ICLS}} = \sqrt{\hat{\sigma}^2 \text{diag}\left[\mathbf{M}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{M}'\right]}, \tag{3.10}$$

where

$$\hat{\sigma}^2 = \left[(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{OLS}})'(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{OLS}})\right] / (n - T), \tag{3.11}$$

and

$$\mathbf{M} = \mathbf{I} + \text{diag}[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{A}'\tfrac{1}{2}\boldsymbol{\lambda}_{\text{KT}}][\text{diag}(\hat{\boldsymbol{\eta}}_{\text{OLS}})]^{-1}. \tag{3.12}$$

If the model is unconstrained, the estimated variance-covariance matrix reduces to the variance-covariance matrix of the ordinary least squares (OLS) estimator. For more details the reader is referred to Liew (1976), Wolak (1987), and Frank & Heiser (in press *a*). The standard errors for the ICLS estimator can be used to construct 95% *t*-confidence intervals in the usual way.

When the tree topology is not known yet and has to be estimated from the sample data first, the theory described in the previous paragraph cannot be applied directly. The standard errors cannot be estimated on the same data that were used to obtain the tree topology. In practice, the problem can be circumvented by dividing the sample in a training set and a test set. The training set is used to derive the tree topology, which is fitted on the test data to obtain the standard errors and the 95% *t*-confidence intervals for the feature discriminability parameters. The rationale behind this approach is the following: assuming that the sample is an adequate representation of the population, the training set will yield a tree topology that is close to the population tree or feature set. The deviations from the true tree topology are assumed to result from sampling error, and, therefore, will probably lead to near zero feature discriminability values and confidence intervals that contain the value zero. These assumptions have been verified by Monte Carlo simulation and the results are provided in the following.

Testing the appropriateness of imposing constraints

In the previous it has been assumed that there exists a representation of the data in terms of (positive) distances between points in a network or a tree. The validity of this assumption can be verified in a hypothesis-testing framework: we can test whether the data is consistent with true values of the parameters satisfying the restrictions imposed on the estimated coefficients. The null hypothesis of the inequality constraints $\mathbf{A}\hat{\boldsymbol{\eta}}_{\text{ICLS}} \geq \mathbf{r}$ (the ICLS solution) can be tested against an unrestricted alternative $\hat{\boldsymbol{\eta}}_{\text{OLS}} \in \mathbf{A}^t$ (the OLS solution). These multivariate inequality constraints lead to the following likelihood ratio test:

$$-2ln\left(\frac{L_{\text{ICLS}}}{L_{\text{OLS}}}\right) = 2(lnL_{\text{OLS}} - lnL_{\text{ICLS}}), \tag{3.13}$$

where L_{ICLS} and L_{OLS} are the maximum values of the likelihood function under the null hypothesis $\mathbf{A}\boldsymbol{\eta} \geq \boldsymbol{r}$ and the alternative hypothesis $\boldsymbol{\eta} \in \mathbf{A}^t$, repectively. If σ^2 is known the *LR* statistic takes the following form:

$$LR = \left[(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{ICLS}})'(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{ICLS}}) - (\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{OLS}})'(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{OLS}})\right] / \sigma^2. \tag{3.14}$$

According to Wolak (1987) the *LR* statistic is also the optimal value of the objective function, or the primal function of the following quadratic programming problem:

$$\min_{\boldsymbol{\eta}} = \left[(\boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta})'(\boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta}) - (\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{OLS}})'(\boldsymbol{\delta} - \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{OLS}})\right] / \sigma^2$$
$$\text{subject to } -\mathbf{A}\boldsymbol{\eta} \geq \mathbf{r}. \tag{3.15}$$

Wolak (1987) showed that the Kuhn-Tucker test statistic (*KT*) is equal to the *LR* test statistic using the theory of quadratic programming, which states that the optimal

value of the objective function of the primal equals that same value for the dual problem under certain conditions. The necessary conditions are that $\mathbf{X}'\mathbf{X}$ is non-singular and $\mathbf{A}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{A}'$ is positive definite. The Kuhn-Tucker test statistic is the optimal value of the dual problem of the objective function of Equation 3.13, and can be formulated as follows:

$$KT = \left[\boldsymbol{\lambda}'_{\mathrm{KT}}\mathbf{A}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{A}'\boldsymbol{\lambda}_{\mathrm{KT}}\right] / 4\sigma^2 \tag{3.16}$$

Wolak (1987) also showed that the KT and the LR statistics have the same distributions and continue to possess the same distribution if the same estimate for σ^2 is used when σ^2 is unknown and replaced by its estimated value $\hat{\sigma}^2$. The null distribution of both test statistics is a weighted sum of Snedecor's F distributions, a property that also holds for covariance matrices other than $\sigma^2\mathbf{I}$. For the hypothesis testing problem $\mathbf{H}_0 : \boldsymbol{\lambda}_{\mathrm{KT}} = 0$ versus $\mathbf{H}_1 : \boldsymbol{\lambda}_{\mathrm{KT}} \geq 0$ (which is equivalent to the testing problem $\mathbf{H}_0 : \mathbf{A}\boldsymbol{\eta} \geq \mathbf{r}$ versus $\mathbf{H}_1 : \boldsymbol{\eta} \in \mathbf{A}^T$), the null distribution of the KT statistic (and the LR statistic) with σ^2 replaced by $\hat{\sigma}^2$ (Equation 3.11), is equal to:

$$\begin{aligned} Pr_{0,4\hat{\sigma}^2\boldsymbol{\Lambda}}\left[KT \geq q\right] &= \sum_{c=1}^{C} Pr[\mathrm{F}_{c,n-T} \geq \tfrac{q}{C}]w(C,c,4\boldsymbol{\Lambda}) \\ Pr_{0,4\sigma^2\boldsymbol{\Lambda}}\left[KT = 0\right] &= w(C,0,4\boldsymbol{\Lambda}), \end{aligned} \tag{3.17}$$

where $\boldsymbol{\Lambda} = (\mathbf{A}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{A}')^{-1}$, and q is the value of the Kuhn-Tucker test statistic. The weights w denote the proportion of times $\boldsymbol{\lambda}_{\mathrm{KT}}$ (Equation 3.16) has exactly c elements larger than zero and can be calculated in closed form for the cases in which $C \leq 4$ (see Wolak, 1987, Appendix). For the cases where the number of constraints exceeds the number 4, Monte Carlo techniques can be used, as will be explained in the Method section.

Estimating prediction error

In addition to the the Kuhn-Tucker test that requires one of the models to be nested within the other (a constrained model versus an unconstrained model), there is an easy way to evaluate the goodness-of-fit of models that are not necessarily nested within each other. Likelihood ratio tests are not suited for testing nonnested models, which have the same number of effective parmeters (Felsenstein, 2004, pp. 316-318; Huelsenbeck & Rannala, 1997). Therefore, Felsenstein (1985, 2004) evaluates the goodness of fit of the tree topology by constructing a consensus tree using a resampling strategy (the nonparametric bootstrap). The AIC statistic (Akaike, 1974) can be used for any pair of models whether nested or not, and has been used for that purpose in phylogenetics (Kishino & Hasegawa, 1990), but also in several MDS applications (Takane, 1981, 1983; Takane & Carroll, 1981; Winsberg & Ramsay, 1981) and is mainly suitable when a log-likelihood loss function is used. Here, we propose a criterion closely related to AIC that is frequently used in the context of linear models: the *generalized cross-validation* (GCV). This statistic provides a convenient approximation to leave-one-out cross-validation for linear fitting under squared-error

loss (Hastie, Tibshirani, & Friedman, 2001, p. 216). A linear fitting method is one for which we can write:

$$\hat{\boldsymbol{y}} = \mathbf{S}\boldsymbol{y}. \tag{3.18}$$

The hat matrix $\mathbf{S}$ from Equation 3.18 is equal to the combination of matrices that transforms the observed data $\mathbf{y}$ into the predicted values $\hat{\mathbf{y}}$.

Using the hat matrix, linear fitting methods can be written as follows,

$$\frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{y}_i]^2 = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{y_i - \hat{y}_i}{1 - S_{ii}}\right]^2, \tag{3.19}$$

where S_{ii} is the ith diagonal element of $\mathbf{S}$. The GCV approximation is

$$GCV = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{y_i - \hat{f}(x_i)}{1 - trace(\mathbf{S})/N}\right]^2, \tag{3.20}$$

where the quantity $trace(\mathbf{S})$ is the effective number of parameters. Applied to the additive tree the *generalized cross-validation* statistic can be computed as follows. From Liew (1976) we know that the following relation exists between the ICLS and the OLS estimator, which leads to the matrices needed to construct the hat matrix:

$$\begin{aligned} \hat{\boldsymbol{\eta}}_{\text{ICLS}} &= \mathbf{M}\hat{\boldsymbol{\eta}}_{\text{OLS}} \\ &= \mathbf{M}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\delta}. \end{aligned} \tag{3.21}$$

From the relation expressed in Equation 3.21 it follows that the predicted distance values can be obtained with:

$$\hat{\mathbf{d}} = \mathbf{X}\mathbf{M}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\delta}, \tag{3.22}$$

and, consequently, the hat matrix is equal to

$$\mathbf{S} = \mathbf{X}\mathbf{M}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'. \tag{3.23}$$

The *generalized cross-validation* error for the additive tree can be estimated using the trace of the hat matrix from Equation 3.23:

$$GCV_{\text{FNM}} = \frac{1}{n}\sum_{l=1}^{n}\left[\frac{\delta_l - \hat{d}}{1 - trace(\mathbf{S})/n}\right]^2. \tag{3.24}$$

3.5 Method Monte Carlo simulations

To evaluate the performance of the statistical inference theory described in the previous section, three Monte Carlo simulations were conducted using data structures that approximate the practice of data analysis with additive tree models. The first simulation shows how to obtain the empirical p-value for the Kuhn-Tucker test described in Equations 3.16 and 3.17. The second simulation study evaluates the performance of the nominal standard errors for known tree topologies compared to

empirical (bootstrap) standard errors. The third simulation study assesses the performance of the nominal standard errors when the tree topology is unknown. In this study the performance of the GCV_{FNM} statistic that serves as an approximation for the prediction error is evaluated as well. All simulation procedures were programmed in Matlab and made use of its pseudo-random number generator, which was set to 1.0 prior to the simulation process.

Empirical *p*-value Kuhn-Tucker test

The null distribution of the Kuhn-Tucker test was calculated by simulating many data sets from a fixed population distribution. The model parameters were estimated from the original data (the *kinship* data) under the null hypothesis, i.e. the inequality constrained least squares model with associated feature parameters as displayed in Table 3.2. A number of 1,000 multivariate normal samples of $n = 105$ dissimilarities were sampled using the binomial distribution to ensure positive dissimilarity values that follow a normal distribution. The details of the method of sampling from the binomial distribution are described in Frank & Heiser (in press *a*). The Kuhn-Tucker test statistic (Equation 3.16) was calculated for each data set, and the proportion of the replicates in which the value of the test statistic exceeded the value obtained for the original data represents the significance level of the test.

Simulation for nominal standard errors with a priori tree topology

The purpose of this simulation study is to evaluate the performance of the nominal standard errors of the ICLS estimator compared to empirical (bootstrap) standard errors, for the situation where the tree topology is known in advance. In addition, the performance of these nominal standard errors are evaluated by comparing the coverage of the nominal confidence intervals with the coverage of bootstrap confidence intervals . The coverage is equal to the proportion of times the true value is included in the confidence interval.

In this simulation study the performance of the standard errors of the ICLS estimator was evaluated using positive true feature parameters, which represents a situation where it is correct to apply constraints and consequently, the asymptotic properties of the ICLS estimator are expected to hold. For the asymptotic properties to hold, normally distributed errors and homogeneous variances are required as well. Given positive true feature parameters, true distances can be computed that can be used as population values from which dissimilarities can be sampled by adding some error to the true distances. True distances were computed with:

$$\mathbf{d} = \mathbf{X}\boldsymbol{\eta}, \tag{3.25}$$

where the true parameters are equal to the ICLS estimates ($\hat{\eta}_{\text{ICLS}}$) in Table 3.2 and $\mathbf{X}$ is obtained with the feature matrix of the *kinship* data (Figure 3.5). The true tree is starlike because several branches have branch lengths equal to zero. A number of $S = 1,000$ samples of $n = 105$ dissimilarities each, was created by sampling from the binomial distribution and with a homogeneous variance condition created with error variance σ^2 equal to 14.4, which corresponds to the observed residual error

variance after fitting the FNM on the original *kinship* data (see for details on the method of binomial sampling, Frank & Heiser, in press *a*). Each simulation sample formed the starting point for a bootstrap of $B = 10,000$ bootstrap samples, using the method of multivariate sampling, which means that for each dissimilarity δ_l ($l = 1, \cdots, n$) sampled from the *kinship* data, the corresponding row of the original $\mathbf{X}$ matrix with features was sampled as well. The simulation yielded 1,000 nominal standard errors ($\hat{\sigma}_{\text{ICLS}}$) for the ICLS estimator. The 1,000 bootstraps (each based on 10,000 bootstrap samples) resulted in 1,000 bootstrap standard deviations (sd_B) of the ICLS estimator.

To evaluate the performance of the estimators, two commonly used measures, the bias and the root mean squared error (*rmse*), were used. Estimates of bias were calculated for the feature parameter estimates $\hat{\eta}_{\text{ICLS}}$, the nominal standard errors $\hat{\sigma}_{\text{ICLS}}$, and the bootstrap standard deviations sd_B. Bias is equal to the expected value of a statistic, $E(\hat{\theta})$, minus the true value θ. For example, the bias of each nominal standard error $\hat{\sigma}_{\text{ICLS}}$ is determined in the simulation study by:

$$bias_{\hat{\sigma}_{\text{ICLS}}} = \left[\frac{1}{S}\sum_{a=1}^{S}\hat{\sigma}_{\text{ICLS}}\right] - \sigma_{\eta}, \tag{3.26}$$

where S indicates the number of simulation samples. The bias of $\hat{\sigma}_{\text{ICLS}}$ is computed with σ_{ICLS} equal to Equation 3.10, using the true values σ^2, $\mathbf{X}$ and $\mathbf{M}$ from Equation 3.12. The bias for the bootstrap standard errors is calculated in the same way, with the exception that $\frac{1}{S}\sum_{a=1}^{S}\hat{\sigma}_{\text{ICLS}}$ is replaced by the sum of the bootstrap standard deviations sd_B.

The *rmse* is equal to the square root of $E[(\hat{\theta} - \theta)^2]$ and takes into account both bias and standard error of an estimate, as can be deduced from the following decomposition (Efron & Tibshirani, 1998):

$$rmse_{\theta} = \sqrt{sd^2_{\hat{\theta}} + bias^2_{\hat{\theta}}}. \tag{3.27}$$

The nominal standard errors ($\hat{\sigma}_{\text{ICLS}}$) were used for the construction of nominal 95% confidence intervals , based on the t distribution ($df = n - T$, with n equal to the number of dissimilarities and T equal to the number of features). Empirical 95% confidence intervals were obtained with the bootstrap-t interval, which is computed in the same way as the nominal confidence interval with the only difference that the bootstrap standard errors (sd_B) are used instead of the estimated standard errors for the sample. For both nominal and empirical confidence intervals , the coverage percentage is equal to the proportion of the simulated samples in which the confidence interval includes the true parameter value.

In a previous study (Frank & Heiser, in press *a*) we also used the *bias-corrected and accelerated* bootstrap interval, the BC_a (Efron & Tibshirani, 1998) in addition to the bootstrap-t interval. Due to the disappointing results obtained for the BC_a intervals, especially when larger numbers of constraints are activated, we restricted this study to the bootstrap-t intervals, which performed better.

Simulation for nominal standard errors with unknown tree topology

The data structure used for this simulation study is based on data from Tversky and Hutchinson (1986, Table 1, p. 5). The data represent mean ratings of similarity between 20 common fruits on a 5-point scale (range 0 - 4, with 4 meaning highly related). For use with an additive tree model, the data were first transformed to dissimilarity values by subtracting each original similarity value from 4. An additive tree was inferred from these dissimilarities with the neighbor-joining (NJ) method (Saitou & Nei, 1987) using the NJ algorithm programmed for Matlab by Strauss (see http://www.biol.ttu.edu/Strauss /Matlab/Matlab.htm). Next, the feature structure (features and internal nodes) was derived from the NJ tree topology by constructing the feature matrix $\mathbf{E}_T$ as in Equation 3.5. The feature matrix equal to the NJ tree topology was submitted to the FNM program (PROXGRAPH) to obtain the ICLS estimates ($\hat{\eta}_{\text{ICLS}}$) for the feature discriminability parameters and the estimated distances. Figure 3.7 shows the resulting tree, where three major clusters become apparent. There is a large cluster with the following three subclusters tropical (exotic) fruit (*coconut, pineapple, pomegranate, banana*), melons (*honeydew, watermelon*) and citrus fruit (*lemon, orange, grapefruit*). This cluster also comprises the tomato that is in a sense exotic because it is not generally recognized as fruit. The second cluster (*grapes, blueberry,*

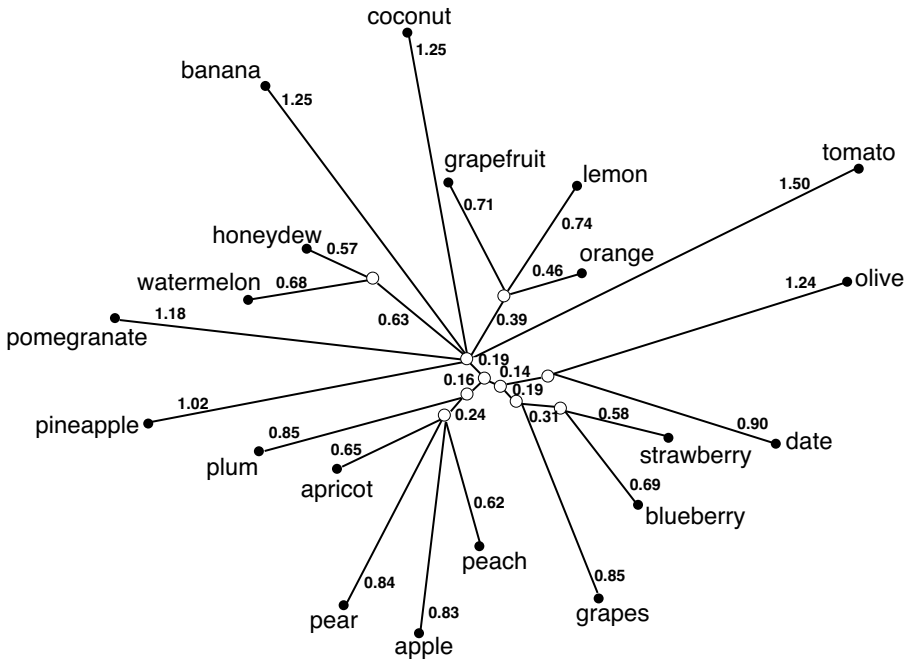

Figure 3.7: Additive tree representation of the *fruit* data obtained with PROXGRAPH based on the tree topology resulting from the neighbor-joining algorithm.

strawberry, date, olive) seems to be determined by the shape and the size of the fruits: small and berry shaped. The third cluster (*plum, apricot, pear, apple, peach*) contains two subfamilies from the rosaceae family, the pome fruits (*pear, apple*) and the stone fruits (*plum, apricot, peach*). The feature structure and the feature discriminability parameters of this tree serve as the true model for the simulation study and are displayed in Table 3.3.

A number of 100 simulation samples with dissimilarities were sampled from the true distances using the aforementioned method of binomial sampling. Two levels of error variance ($\sigma^2 = 0.5, \sigma^2 = 1.0$) were used. To obtain the nominal standard errors when the tree topology is unknown, each sample was divided in a training set and a test set such that the test set contained a proportion of 0.33 of the total sample size. Three levels of total sample size were used: 50, 100 and 300 observations. A total sample size of 50 means that 50 subjects evaluated the relatedness of the 20 fruits on a 5-point scale. The test set contains 33% of the total sample size and the training set the remaining observations. The data that were analyzed were the mean values of the total dissimilarity values in the training set and in the test set. The mean dissimilarity values of the training set of each simulation sample were submitted to the NJ algorithm to obtain an NJ tree topology. Next, the feature structure (features and internal nodes) was derived from this tree topology by constructing the feature matrix $\mathbf{E}_T$ as in Equation 3.5. The feature parameters and associated nominal standard errors were obtained by fitting the training tree topology on the test set dissimilarities using PROXGRAPH. In addition, the prediction error of each sample was estimated with the GCV_{FNM} statistic (Equation 3.24), which was estimated for each test sample using the tree topology obtained in the training sample. The same GCV_{FNM} statistic was also estimated with the training tree topologies and the true distances instead of the test sample dissimilarities. With no sampling error present, the GCV_{FNM} values give an unbiased estimate of the error due to model misspecification. Both GCV_{FNM} estimates were compared in all experimental conditions. The performance of the GCV_{FNM} statistic was further assessed by comparing its distribution in the 6 experimental conditions to the distribution of the number of true features that were recovered in the training tree topologies. Tree topologies that recover a large number of true features should have lower estimates of prediction error.

The performance of the nominal standard errors ($\hat{\sigma}_{\text{ICLS}}$) was evaluated by the coverage proportions of t-confidence intervals constructed with estimates of the nominal standard errors ($\hat{\sigma}_{\text{ICLS}}$). The coverage percentage is equal to the proportion of the simulated samples in which the confidence interval includes the true feature discriminability value, in the same way as for the simulation with fixed tree topology. There is, however, an important difference, because, in this simulation study, an NJ tree topology was estimated for each simulation sample. As a result, the training sample of each simulation sample yielded a feature set that does not necessarily contain all the features present in the true tree topology. Therefore, the proportion of confidence intervals that include the true feature discriminability value can only be obtained for feature discriminability parameters associated with features that are part of the true tree topology. In practice, this means that each tree topology inferred for the training samples, was compared to the true tree topology and only

the nominal standard errors associated with features that belong to the true model were used to obtain the coverage proportions of the t-confidence intervals . The feature discriminability parameters ($\hat{\eta}_{\text{ICLS}}$) belonging to features that are not included in the true tree topology were also evaluated. To verify the assumption that features that are not included in the true model will lead to small $\hat{\eta}_{\text{ICLS}}$ values, t-confidence intervals were constructed using the nominal standard errors ($\hat{\sigma}_{\text{ICLS}}$). The proportion of the confidence intervals that contain the value zero provided evidence for the tenability of the aforementioned assumption.

A few words have to be said about the method used to compare the features resulting from the training sample tree topology with the features from the true tree topology. In terms of a feature model, the tree topology consists of a set of features that are binary $(0, 1)$ variables. A binary vector is in fact the binary code representation of an integer. In the same way, features can be considered as unique representations of integers with the number of bits equal to the number of objects (m). Although there are several binary coding systems available, the Gray code system was used because in the context of FNM it proved to be an efficient method to generate the complete set of distinctive features (Frank & Heiser, in press b). In this simulation study, the Gray code system was used to derive the unique Gray code rank number for the features in the true tree topology and for the features in the training sample topologies. The Gray code rank numbers were derived using a Matlab transcription by Burkardt (see http://www.csit.fsu.edu/ burkardt/) of the original algorithms for generating Gray codes in Nijenhuis and Wilf (1978). Since the binary feature vectors can be uniquely identified by a Gray code rank number, the comparison between the features of the training tree topologies and the features of the true tree topology amounts to a simple comparison of integers.

Table 3.3: The 17 cluster features (F_1 - F_{17}) and 20 unique features (F_{18} - F_{37}) with associated feature discriminability parameters for the neighbor-joining tree on the *fruit* data.

Feature	Objects	$\hat{\eta}_{\text{ICLS}}$
F_1	watermelon, honeydew	0.634
F_2	strawberry, blueberry	0.309
F_3	orange, lemon	0.116
F_4	orange, grapefruit, lemon	0.386
F_5	date, olive	0.300
F_6	grapes, strawberry, blueberry	0.192
F_7	pineapple, coconut	0.159
F_8	apple, pear	0.102
F_9	peach, apricot	0.245
F_{10}	peach, apricot, plum	0.038
F_{11}	grapes, strawberry, blueberry, date, olive	0.136
F_{12}	orange, grapefruit, lemon, watermelon, honeydew	0.004
F_{13}	F_{12} + pineapple, coconut	0.095
F_{14}	F_{13} + pomegranate	0.063
F_{15}	apple, peach, pear, apricot, plum	0.155
F_{16}	F_{14} + tomato	0.019
F_{17}	F_{16} + F_{11}	0.009
F_{18}	orange	0.461
F_{19}	apple	0.832
F_{20}	banana	1.253
F_{21}	peach	0.615
F_{22}	pear	0.838
F_{23}	apricot	0.645
F_{24}	plum	0.850
F_{25}	grapes	0.846
F_{26}	strawberry	0.576
F_{27}	grapefruit	0.709
F_{28}	pineapple	1.023
F_{29}	blueberry	0.694
F_{30}	watermelon	0.682
F_{31}	honeydew	0.568
F_{32}	pomegranate	1.179
F_{33}	date	0.895
F_{34}	coconut	1.247
F_{35}	tomato	1.506
F_{36}	olive	1.235
F_{37}	lemon	0.739

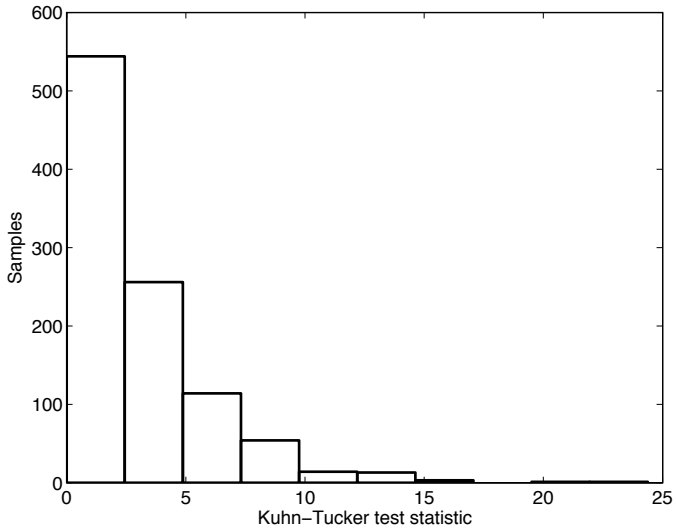

Figure 3.8: Histogram of Kuhn-Tucker test statistic obtained with parametric bootstrap (1,000 samples) with ICLS as H_0 model, based on *kinship* data. The empirical p-value is equal to .74 and represents the proportion of samples with values on the Kuhn-Tucker statistic larger than 0.89, the value of the statistic observed for the sample.

3.6 Results simulation

Results Kuhn-Tucker test and estimates of prediction error

Figure 3.8 shows the result of the simulation based on the additive tree model obtained on the *kinship* data. The Kuhn-Tucker test statistic for the original sample is equal to 0.89 and a proportion of 0.74 of the 1,000 simulated samples have values equal or larger to the sample value of the statistic under the H_0. Therefore, there is no reason to reject the null hypothesis and consequently, it seems appropriate to apply the positivity constraints on these data.

Concerning the estimates of prediction error, the resolved tree yields a GCV_{FNM} value equal to 278.37 and the unresolved tree has $GCV_{\text{FNM}} = 246.10$. Only relative magnitudes of this statistic are meaningful and the conclusion is that the unresolved tree has less prediction error. In summary, the result of the Kuhn-Tucker test shows that the inequality constraints reasonably fit the data, and the estimate of prediction error shows that the unresolved tree has better prediction properties.

Performance of the nominal standard errors for known tree topology

Figure 3.9 shows the mean, the bias, and the *rmse* of the distribution of the 1,000 nominal standard errors as well as the distribution of the 1,000 bootstrap standard errors plotted against the true variability values. Plotting against the true variability

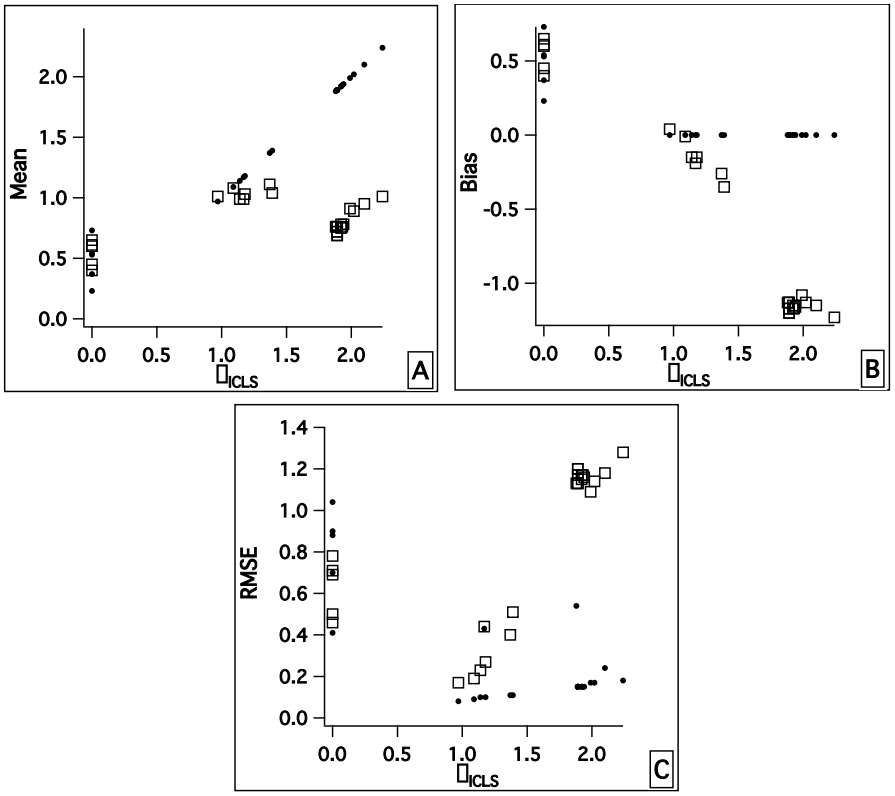

Figure 3.9: Mean (panel A), bias (panel B), and *rmse* (panel C) of the 1,000 simulated nominal standard errors $\hat{\sigma}_{\text{ICLS}}$ (•) and the 1,000 bootstrap standard deviations sd_B (□) plotted against the true nominal standard errors σ_{ICLS}.

allows for comparing the results for the parameters with activated constraints (nominal standard errors equal to zero) and the remaining parameters with no activated constraints. The distribution of the bootstrap standard deviations and the nominal standard errors show a different pattern depending whether constraints are activated or not. When constraints are activated, the pattern of the nominal standard errors is almost equal to the pattern of the bootstrap standard deviations: the values of the mean (panel A), the bias (panel B) and the *rmse* (panel C) are very related. When constraints are not activated, the distribution of the bootstrap standard deviations reveals a clearly different pattern compared to the nominal standard errors. The mean of the bootstrap standard deviations (panel A) is evidently smaller than the mean of the nominal standard errors, which are very close to the true variability

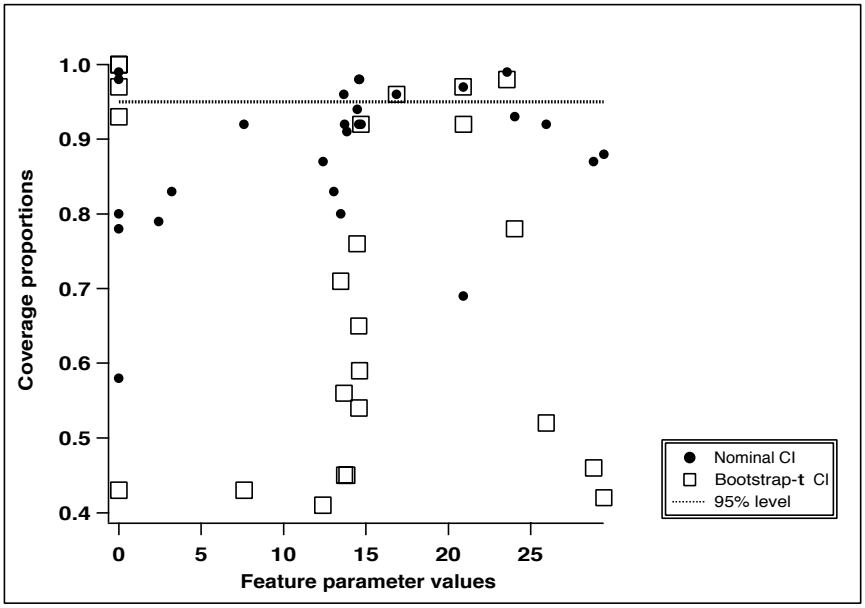

Figure 3.10: Coverage proportions of the nominal *t*-CI and bootstrap *t*-CI for the true feature discriminability values, based on the 1,000 simulated samples.

values, and, consequently, the bootstrap standard deviations are biased downwards (panel B), showing an underestimation of the true variability, whereas the nominal standard errors show almost no bias. The larger bias values for the bootstrap standard deviations, combined with larger variability (not shown) lead to larger values for the *rmse* (panel C).

Figure 3.10 shows the coverage proportions of the nominal and the bootstrap 95% *t*-confidence intervals for the ICLS estimator. The coverage of the nominal confidence intervals is closer to the nominal 95% level than the coverage of the bootstrap confidence intervals that are mostly lower than the nominal values and achieve several low coverage values around 40%. This finding corresponds with the patterns observed in Figure 3.9, where the bootstrap standard deviations are clearly biased downwards.

Performance of the nominal standard errors for unknown tree topology

The right panel of Figure 3.11 displays the distribution of the number of true cluster features present in the NJ tree topologies inferred for the 100 training samples in each experimental condition. Since the unique features are always the same for each topology, only the cluster features are represented. There are 17 cluster features in the true tree topology for the simulation study. The NJ tree topologies obtained in the training samples consistently had 17 cluster features, with two exceptions only. In

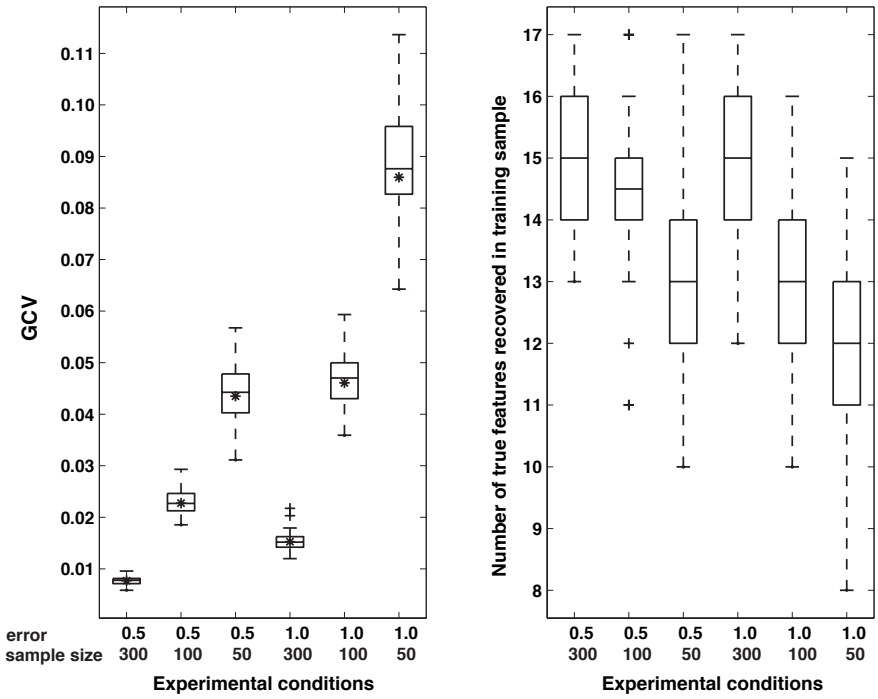

Figure 3.11: *Left panel*: Distribution of the GCV_{FNM} statistic estimated on the test samples based on the tree topology inferred for the training samples under all experimental conditions for 100 simulation samples. The asterisk in each box represents the mean of the true GCV_{FNM} values. *Right panel*: Distribution of the number of cluster features equal to the true cluster features ($T_C = 17$) present in the tree topologies obtained for the training samples of the same 100 simulation samples in each experimental condition.

the low error condition with sample size 100 and with sample size 50, 1 sample out of 100 yielded a NJ tree topology with 16 cluster features. The boxplots in the right panel of Figure 3.11 show that the number of true features recovered in the training sample decreases when the sample size decreases and when the error level is higher, except for sample size 300. The distributions of the prediction error, estimated with the GCV_{FNM} statistic on each test sample (left panel of Figure 3.11), mirror these effects: higher levels of GCV_{FNM} correspond to less well recovered tree topologies. To evaluate the performance of the GCV_{FNM}, this statistic was also estimated with the training tree topology fitted on the true distances. The mean of these GCV_{FNM} values in each of the experimental conditions is represented with an asterisk in the left panel of Figure 3.11 and it is clear that the mean of the GCV_{FNM} values in the test samples is very close to the mean of the GCV_{FNM} values obtained for the true distances.

Table 3.4: Proportion of 95% *t*-confidence intervals containing the value zero in the test samples for the feature discriminability parameters associated with features not present in the true tree topology

Error level	0.5	0.5	0.5	1.0	1.0	1.0
Sample size	300	100	50	300	100	50
Coverage						
1.00	1.00	0.82	0.85	0.92	0.80	0.81
0.99	0.00	0.18	0.15	0.00	0.15	0.14
0.98	0.00	0.00	0.00	0.08	0.04	0.03
0.96	0.00	0.00	0.00	0.00	0.00	0.02

The feature discriminability values for the features in the training samples that are not included in the true model were recorded for all experimental conditions. Most of these feature discriminability values were equal to zero, but some reached higher values, with a maximum value of 0.26. However, most of these values did not significantly differ from zero, as can be deduced from the coverage proportion of the *t*-intervals in Table 3.4. In general, at least 96% of the confidence intervals contained the value zero. When error level was low and sample size was equal to 300, all confidence intervals contained the value zero. With increasing error level and decreasing sample sizes, the proportion of confidence intervals spanning zero gradually drops off to 0.96. These results lead to the conclusion that, in general, the feature discriminability parameters associated with features that are not part of the true tree topology, had values that do not significantly differ from zero. Even in the worst case, only a very small proportion (4%) of the feature discriminability parameters associated with features that are not part of the the true tree topology, had values that differ significantly from zero.

Figure 3.12 gives insight in the performance of the nominal standard errors in each experimental condition related to the proportion of correctly recovered features in the training samples. The squares indicate the proportion of features in the NJ tree topologies inferred for the training samples that correspond to the features in the true tree topology. The set of unique features (corresponding to the numbers 18 to 37 in Figure 3.12) is by definition part of the tree topology and therefore, these features have perfect recovery results in all experimental conditions. The recovery of the cluster features is clearly affected by the experimental conditions. The set of features that are less well recovered form the following subset $\{F_3, F_7, F_8, F_{10}, F_{12}, F_{13}, F_{14}, F_{16}, F_{17}\}$. When sample size decreases and error becomes higher, an increasing number of features from this set are less well recovered. It is, however, not surprising that this particular set of features is not well recovered because these features have the smallest feature discriminability parameters in the total feature set (see, Table 3.3). From the point of view of interpretation, these less well recovered cluster features form subsets of fruits that are counterintuitive, like, for example, the combination of citrus fruits and the two types of melons, represented by F_{12} (Table 3.3).

The bullets in Figure 3.12 represent the proportion of nominal t-confidence intervals in the test samples that cover the true feature discriminability parameter for the features that are part of the true tree topology. The feature discriminability parameters that have lower coverage proportions are associated with the same subset of features that are less well recovered. The coverage proportions of the nominal t-confidence intervals are adequate (ranging from 0.95 to 1.0) for the features that are well recovered, but become lower (sometimes reaching values lower than .40) for the features that are less well recovered.

3.7 Discussion

This paper showed how to obtain theoretical standard errors and confidence intervals for the estimates of branch lengths in psychometric additive trees for a priori known tree topologies as well as for estimated tree topologies. The statistical inference theory proposed here derives from the multiple regression framework, which is directly related to the feature representation of additive trees. Using features along with the univariate multiple regression framework offers a different perspective on statistical inference in psychometric additive trees and might be useful for the phylogenetic tree domain as well.

However, a comparison between evolutionary trees and psychometric trees is not straightforward because different assumptions are made about the estimated distances in the tree, and, consequently, the results might not be exchangeable between the two types of tree models. In phylogenetic trees, the distances in the tree represent evolutionary distances, which in most cases are equal to the number of nucleotide substitutions for all pairs of nucleotide sequences representing the species. In psychometrics there is no generally accepted theory about the underlying distribution of dissimilarities between objects. In multidimensional scaling theory, several possible distributions have been proposed. Ramsay (1982) suggested the normal distribution, the log-normal distribution (because of the nonnegative nature of dissimilarities) and a symmetric alternative, the inverse Gaussian (or Wald) distribution. Restle (1961) proposed the gamma distribution and Takane (1981) and Takane and Carroll (1981) used various distributions that take into account the specific data generation process that underlies each data collection method.

Despite these differences, both types of tree domains share the following important property: from evolutionary perspective, but also in psychology, a tree with negative branch lengths has no meaning and cannot be accurate by definition. Consequently, all tree searching algorithms search for tree topologies with positive branch lengths while discarding all tree topologies that yield negative estimates of branch lengths. Searching for a tree with positive branch lengths implies that positivity constraints should be imposed on the estimates of the branch lengths. Imposing inequality constraints during estimation has consequences for the statistical properties of the estimates: they become biased because their distribution is truncated at zero. The presence of the inequality constraints cannot be ignored and should be part of the tree searching algorithms, as already pointed out by Gascuel and Levy (1996), but also when the variability of the branch lengths are estimated.

This paper shows that the theoretical standard errors for inequality constrained least squares estimates are useful in assessing the variability of the branch lengths in psychometric additive trees. For a priori known tree topologies the theoretical standard errors perform well. When the tree topology is not known in advance and estimated with the NJ method, the performance of the confidence intervals based on the theoretical standard errors is adequate, except for the features that have very small feature discriminability values and, at the same time, are not well recovered by the NJ method.

The results of this study are however limited to the normal distribution assumption, necessary in the inequality constrained least squares framework. In addition, the assumption of homogeneous variances (Equation 3.7) is arguable because the dissimilarity values that share the same objects are likely to be correlated. In multidimensional scaling, solutions have been proposed by Ramsay (1982) who uses a multiplicative variance components model instead of the additive model of Equation 3.7 and introduces MINQUE variance estimates for cases where the configuration matrix is known. In the phylogenetic domain, Bulmer (1991) uses generalized least squares to account for the heterogeneity. In theory, the combination of ICLS estimates with generalized least squares, yielding inequality constrained generalized least squares estimates (ICGLS), could be a solution. In practice, the statistical properties are barely known (Werner, 1990; Werner & Yapar, 1996) and since the generalized least squares estimates become a computational burden with large number of objects (*cf.* Felsenstein, 2004), the ICGLS estimates are a difficult way to go.

The use of features as predictor variables in a multiple regression framework has additional advantages. An important advantage is that measures of prediction error that are regularly used in this framework become easily available. In this paper we showed how the statistical inference theory of the inequality constrained least squares estimator can be incorporated in the general theory of linear fitting methods to obtain an estimate of prediction error, the *generalized cross-validation* statistic, which is a convenient closed form formula that approximates leave-one-out cross-validation. Besides the very low computational costs, another advantage of the *generalized cross-validation* statistic is that it can be used to compare different tree topologies with the same number of degrees of freedom, i.e. models that have the same number of predictors or features. For the likelihood ratio test, a commonly used test to compare phylogenies, the comparison of tree topologies with the same degrees of freedom is a problem because the test is limited to the case of nested topologies (*cf.* Felsenstein, 2004, Chapter 19).

Another advantage of considering the feature framework for additive trees, is that a frequently used test in the phylogeny domain, testing speciation or population splitting, can be done explicitly by adding *cluster features* to the model. In phylogenetic trees, internal nodes are usually called branching points and indicate that an important event of speciation or population splitting occurred there (*cf.* Nei et al., 1985). The internodal distances are not observed and therefore are inferred form the other, non-internodal distances, as well as the associated standard errors. Several tests for the branching points have been proposed (Bulmer, 1991; Li, 1989; Nei et al., 1985; Tajima, 1992). An alternative for the interior-branch test is the bootstrap method proposed by Felsenstein (1985), which calculates the proportion of bootstrap

trees that agree with the original tree topology inferred for the sample. A population splitting that occurs in a large proportion of bootstrap trees is considered to be very plausible. Sitnikova, Rzhetsky, and Nei (1995) compared the interior-branch test with the bootstrap test and concluded that the bootstrap test tends to yield conservative confidence values compared to the interior-branch test and that the difference between the two tests becomes more salient when the true tree is starlike, which means that some branches have length zero resulting from the correction of negative branch lengths.

Considering features in additive trees allows for a different way to test the branching points. In the phylogeny literature, the branching points result from a certain topology that depends on the tree finding algorithm. FNM offers the possibility to test explicitly for specific ancestral species just by adding cluster features to the feature matrix $\mathbf{E}_T$. The values of the feature discriminability parameters and the associated confidence intervals indicate whether the ancestral species are plausible. The simulation study in this paper for the case of unknown tree topologies, inferred for each simulation sample with the NJ method, is in fact a parametric version of Felsenstein's bootstrap method (1985) because it calculates the proportion of features from the true topology that are recovered in the samples while assuming a model with normally distributed error terms. The confidence intervals obtained with the theoretical standard errors for the feature discriminability parameters led to the same conclusion about the most plausible features (including cluster features that indicate speciation) in the model, but at much less computational cost.

Although strong assumptions have to be made (normally distributed errors and homogeneous variances), we believe that the theoretical standard errors for the inequality constrained least squares estimates are useful for estimating the variability of branch lengths of tree topologies obtained with algorithms like ADDTREE and NJ, which use least squares estimates for the branch lengths. Using features along with the multiple regression framework has many advantages, as has been demonstrated in this paper. Nevertheless, the question remains whether these results are useful for the phylogenetic trees. The answer relies on the challenge to combine the theoretical standard errors for the inequality constrained least squares estimator with the many methods proposed in the phylogenetic literature that take into account the way the evolutionary distances were obtained.

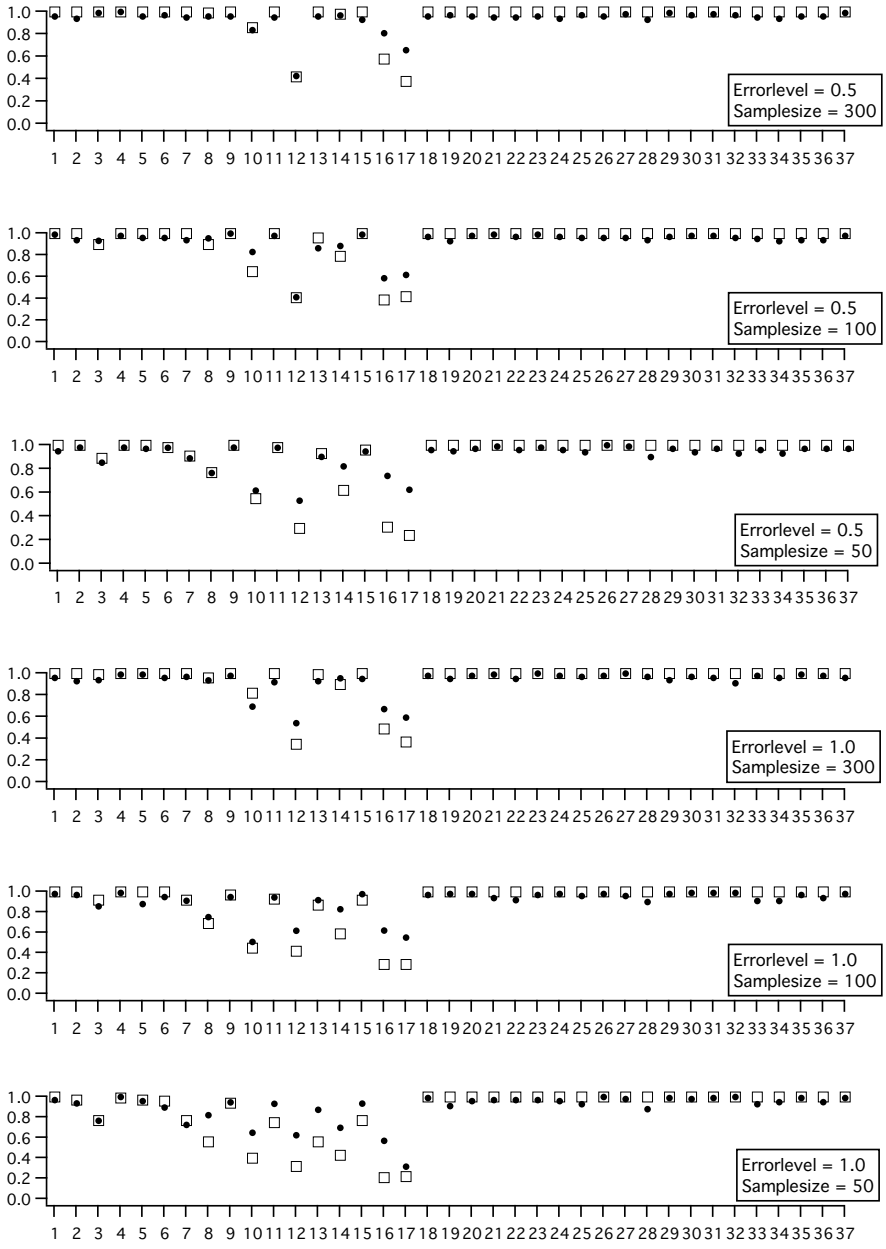

Figure 3.12: Coverage proportions in all experimental conditions for feature discriminability parameters based on nominal t-CI (•) in the test samples and proportions recovered true features in the training samples (□) for each of the 37 features forming the true tree topology.

Chapter 4

Feature Selection in Feature Network Models: Finding Predictive Subsets of Features with the Positive Lasso [1]

Abstract

A set of features is the basis for the network representation of proximity data achieved by Feature Network Models (FNM). Features are binary variables that characterize the objects in an experiment, with some measure of proximity as response variable. Sometimes features are provided by theory and play an important role in the construction of the experimental conditions. In some research settings, the features are not known a priori. This paper shows how to generate features in this situation and how to select an adequate subset of features that takes into account a good compromise between model fit and model complexity, using a new version of Least Angle Regression that restricts coefficients to be nonnegative, called the Positive Lasso. It will be shown that features can be generated efficiently with Gray codes that are naturally linked to the FNM. The model selection strategy makes use of the fact that FNM can be considered as a univariate multiple regression model. A simulation study shows that the proposed strategy leads to satisfactory results if the number of objects $\leqslant 22$. If the number of objects is larger than 22, the number of features selected by our method exceeds the true number of features in some conditions.

4.1 Introduction

Feature Network Models or FNM (Heiser, 1998) are graphical models that represent proximity data in a discrete space while using the same formalism that is the basis of least squares methods used in multidimensional scaling. A typical application area for FNM would be cognitive psychology where one studies how human cognition

[1] This chapter has been accepted for publication as: Frank, L. E. & Heiser, W. J. (in press). Feature selection in Feature Network Models: finding predictive subsets of features with the Positive Lasso. *British Journal of Mathematical and Statistical Psychology.* With an exception for the notes in this chapter and Figure 4.2, which are reactions to remarks made by the members of the promotion committee.

processes stimuli by analyzing the ratings of perceived (dis)similarity of these objects. If N respondents evaluate the dissimilarity of m objects and T binary features characterize these objects, the number of features in which two objects are distinct yields a dissimilarity coefficient that can be used as a structural model to be fitted to the data. The additivity properties of networks make it possible to consider the model as a univariate multiple linear regression problem with positivity restrictions on the parameters. The positivity restrictions are necessary because the parameters represent edge lengths in the network representation of the models.

Least squares and multiple linear regression estimates have been frequently applied in models that are related to FNM, like, for example, extended similarity trees (Corter & Tversky, 1986) and additive clustering with ADCLUS (Shepard & Arabie, 1979) or with MAPCLUS (Arabie & Carroll, 1980). However, the possibilities offered by multiple linear regression have not been fully explored in the context of these models. For instance, statistical inference is not common practice for these clustering and tree models. Recently, theoretical standard errors were introduced and used to construct confidence intervals for the parameters of the FNM (Frank & Heiser, in press a) and related additive trees (Frank & Heiser, 2004) using the theory of nonnegative least squares. In this article, we use the multiple linear regression framework for the selection of a subset of features that constitutes a good compromise between model fit and model complexity.

Before introducing the feature selection strategy proposed in this work, more has to be said about the nature of the features and the feature sets that are used to represent the proximities. The concept of a feature was introduced in psychology by Tversky (1977) who proposed the Contrast Model (CM) to describe the similarity between two objects in terms of a linear combination of the features they share (common features) and the features that distinguish between them (distinctive features). The Contrast Model in its most general form has been used in practice with a priori features only (Gati & Tversky, 1984; Keren & Baggen, 1981), but many models have been developed since, which search for either the common features part or the distinctive features part of the model, or a combination of both. Models based on common features are additive similarity trees (Sattath & Tversky, 1977) and additive clustering (ADCLUS, Shepard & Arabie, 1979; MAPCLUS, Arabie & Carroll, 1980; CLUSTREES, Carroll & Corter, 1995). The distinctive features are used for the extended similarity trees (EXTREE) proposed by Corter and Tversky (1986). A model that has the closest relation to the CM is the Modified Contrast Model (MCM) developed by Navarro and Lee (2004) that aims at finding a set of both common and distinctive features that best describes the data. This model comprises an implementation of Tversky's Contrast Model as well as the Common Features (CF) and the Distinctive Features (DF) models, that are special cases of both CM and MCM. FNM is based on the distinctive features only.

All aforementioned methods aim at finding a set of features that does not necessarily have a nested structure as required in hierarchical trees and additive trees. Rather, a less restricted structure of possibly overlapping clusters or features is sought. The FNM is the only model that represents this overlapping feature structure by a network representation. To find such a feature structure, we propose a strategy that is related to the predictor selection problem in the multiple regression

framework. The basic idea is to generate a very large number of features (or if possible, the complete set of features) first, and then select the best set of features with a subset selection algorithm. We used the Lasso option of the Least Angle Regression (LARS) algorithm (Efron et al., 2004), a recently developed efficient model selection algorithm that is less greedy than the traditional forward selection methods used in the multiple linear regression context, in the sense that the traditional methods have the tendency to eliminate useful predictors that happen to be correlated with the predictor selected in the previous step. We modified the Lasso option of this algorithm into a Positive Lasso to meet the positivity constraints of our model.

The large number of features presented to the subset selection algorithm is generated using the Gray code, a cyclic permuted version of the usual binary code, which will be explained below. Gray codes have a natural link with the network representation of the feature profiles for the objects in the FNM. In addition, the symmetry property of distinctive features leads to a very efficient use of the Gray codes because only half of the total number of codes is sufficient to enumerate the set of all possible distinctive features. This property allows in practice for enumerating the total number of features for numbers of objects m smaller or equal to 22. If m exceeds 22 and complete enumeration is no longer possible, we propose the use of a very large sample of Gray codes combined with a filter technique to reduce the number of features before using subset selection.

The strategy proposed here is different from the algorithms for the other methods because it approaches the problem of finding an adequate set of features from a different angle: most methods search for sets of features while fixing the number of features in advance. Typically, several solutions with different numbers of features are generated, and the best set of features is selected based on criteria such as goodness-of-fit and interpretability. The first application of FNM used a cluster differences scaling algorithm (Heiser, 1998) with number of clusters equal to two, which constitutes a one-dimensional MDS problem with the coordinates restricted to form a bipartition. It is still a hard combinatorial problem, and, therefore the implementation uses a nesting of several random starts together with K-means type of reallocations. The strategy proposed in this paper incorporates model selection criteria during the search process, leading to a set of features that is not necessarily optimal in the current data, but that has predictive value with a balanced trade-off between goodness-of-fit and prediction accuracy. Prediction accuracy or prediction error, which can be assessed with closed form formulas or can be approximated with cross-validation techniques, has not been used yet in this context, except for the Modified Contrast Model (Navarro & Lee, 2004) that uses a forward feature selection method and a model selection criterion related to the BIC criterion.

The remainder of the article is organized as follows. The second section presents the theory of the Feature Network Models and the generation of binary features using Gray codes. The section ends with an application on a data set and a comparison of features provided by theory and features selected by the strategy we propose. The third section shows the results of a simulation study that evaluates the performance of our strategy, and the last section provides concluding remarks.

Table 4.1: Matrix of 16 English consonants, their pronunciation and phonetic features

Consonants		F_1^*	F_2	F_3	F_4	F_5	F_6	F_7
p	(**p**ie)	0	0	0	0	0	1	0
t	(**t**ie)	0	0	0	0	1	0	0
k	(**k**ite)	0	0	0	0	0	0	1
f	(**f**ie)	0	0	1	0	0	1	0
θ	(**th**igh)	0	0	1	0	1	0	0
s	(**s**igh)	0	0	1	1	1	0	0
ʃ	(**sh**y)	0	0	1	1	0	0	1
b	(**b**uy)	1	0	0	0	0	1	0
d	(**d**ie)	1	0	0	0	1	0	0
g	(**g**uy)	1	0	0	0	0	0	1
v	(**v**ie)	1	0	1	0	0	1	0
ð	(**th**y)	1	0	1	0	1	0	0
z	(**Z**ion)	1	0	1	1	1	0	0
ʒ	(vi**s**ion)	1	0	1	1	0	0	1
m	(**m**y)	1	1	0	0	0	1	0
n	(**n**igh)	1	1	0	0	1	0	0

*F_1 = *voicing*; F_2 = *nasality*; F_3 = *affrication*; F_4 = *duration*; F_5 = *place, middle*; F_6 = *place, front*; F_7 = *place, back*.

4.2 Theory

Feature Network Models

Feature Network Models (FNM) are graphical structures that represent proximity data in a discrete space. The properties of these models will be explained using a well known data set, the perceptual confusions among 16 English consonants collected by Miller and Nicely (1955). These 16 phonemes can be described by 7 articulatory features[2]: *voicing*, *nasality*, *affrication*[3], *duration*[4] and three *places of articulation* (see Table 4.1). The authors were particularly interested in which articulatory features are important in distinguishing the consonants when affected by varying signal to noise conditions. The original data consist of 17 matrices in which each cell contains the frequencies of confusion between the spoken phoneme (the rows) and the phoneme written down by the participants (the columns). Shepard (1972) pooled the data from the first six original matrices (representing 6 different signal-

[2]It should be noted that the feature set consists of 7 features instead of the 6 features used for the same data in Chapter 2. The articulatory feature *place of articulation* has three levels (*front, middle, back*) and is represented in Table 4.1 by the three binary features F_5, F_6 and F_7 as a result of dummy coding. Representing the three levels by three variables leads to multicollinearity, and as a result, the third level has been left out from the feature set in Chapter 2. In the present chapter, the technique of the (Positive) Lasso is robust to multicollinearity and therefore, the complete feature set is used.

[3]At present, phonetic experts would call this feature *friction*.

[4]The feature *duration* is not a proper phonetic feature and has been adopted arbitrarily by Miller & Nicely (1955) to distinguish the difference between {s, ʃ, z, ʒ} and the remaining consonants.

to-noise conditions) collected by Miller and Nicely and coverted the pooled data to a symmetric matrix of similarities with the transformation $\varsigma_{ij} = (f_{ij} + f_{ji})/(f_{ii} + f_{jj})$, where f denotes the frequencies of confusion. For our study, the similarities were further transformed into dissimilarities δ_{ij} by the transformation $\delta_{ij} = -\log(\varsigma_{ij})$, assuming that the similarity measures decay exponentially with distance.

The data are illustrative for the use of features provided by theory, i.e., phonetic theory describes the articulatory properties of the phonemes. In many situations, no theory is available about the objects. Features are binary variables indicating for each object whether a particular characteristic is present or absent. Features are not always intrinsically binary: any ordinal or even interval variable if categorised can be transformed into a set of binary features, using dummy coding. For example, the place of articulation has three categories to indicate the place in the mouth where the phonemes are pronounced: front, middle and back. Dummy coding produces the three features *place, front*, *place, middle*, and *place, back* (Table 4.1).

Some set theoretic properties of the binary feature matrix lead to the estimation of a distance measure that approximates the observed dissimilarities. For example, the phoneme g has feature {*voicing, place back*} and phoneme v has the features {*voicing, affrication, place front*}. The difference between the union and the intersection (= the symmetric set difference) expresses which feature g has that v does not have and vice versa: $(g \cup v) - (g \cap v) = \{$*affrication, place front, place back*$\}$. Following Goodman (1951, 1977) and Restle (1959, 1961), a distance measure that satisfies the metric axioms can be expressed as a simple count τ of the elements of the symmetric set difference, a count of the non common elements, between the stimuli O_i and O_j and becomes the *feature distance*: $d(O_i, O_j) = \tau[(O_i \cup O_j) - (O_i \cap O_j)]$.

If $\mathbf{E}$ is a binary matrix of order $m \times T$ that indicates which features t describe the m objects, as in Table 4.1, the re-expression of the feature distance in terms of coordinates is as follows (Heiser, 1998):

$$\begin{aligned} d(O_i, O_j) &= \tau[(O_i \cup O_j) - (O_i \cap O_j)] \\ &= \sum_t |\mathbf{e}_{it} - \mathbf{e}_{jt}|, \end{aligned} \tag{4.1}$$

This re-expression of the feature distance in terms of binary coordinates is also known as the *Hamming distance*. The feature distance used in FNM is a weighted version of the distance in Equation 4.1:

$$d(O_i, O_j) = \sum_t \eta_t |\mathbf{e}_{it} - \mathbf{e}_{jt}|, \tag{4.2}$$

where the weights η_t express the relative contribution of each feature.

If we string out the dissimilarities into a vector, we can use a univariate multiple linear regression model for the dissimilarities:

$$\boldsymbol{\delta} = \mathbf{X}\boldsymbol{\eta} + \boldsymbol{\epsilon}, \tag{4.3}$$

where $\boldsymbol{\delta}$ is a $n \times 1$ vector with dissimilarities, $\mathbf{X}$ is a known $n \times T$ binary $(0, 1)$ matrix of rank T, with n equal to all possible pairs of m objects, i.e., $\frac{1}{2}m(m-1)$, $\boldsymbol{\eta}$ is a $T \times 1$

vector with feature discriminability parameters, and $\boldsymbol{\epsilon}$ is a $T \times 1$ vector. We assume that $\boldsymbol{\epsilon}$ is a $n \times 1$ random vector that follows a normal distribution,

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mu, \sigma^2 \mathbf{I}), \tag{4.4}$$

where $\mathbf{I}$ is an identity matrix of rank n, and where it is assumed that σ^2 is small enough to ensure the occurrence of negative dissimilarities to be negligible. The feature parameters are estimated by minimizing the following nonnegative least squares loss function:

$$\min_{\boldsymbol{\eta}} = \|\boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta}\|^2 \qquad \text{subject to } \boldsymbol{\eta} \geq 0, \tag{4.5}$$

where the feature parameters $\boldsymbol{\eta}$ are constrained to be positive because they represent edge lengths in the network representation of the network, as will be explained in the next paragraph. To be able to express the loss function of the FNM in a more convenient multiple regression problem as done in Equation 4.5, the original matrix $\mathbf{E}$ must be transformed first. The matrix $\mathbf{X}$ is obtained by applying the following transformation on the rows of matrix $\mathbf{E}$ for each pair l of the total of n pairs of objects, where the elements of $\mathbf{X}$ are defined by:

$$x_{lt} = |e_{it} - e_{jt}|, \tag{4.6}$$

where the index $l = 1, \cdots, n$ varies over all pairs (i, j). The result is the binary $(0, 1)$ matrix $\mathbf{X}$, where each row contains the featurewise distances for each pair of objects, with 1 meaning that the feature is distinctive for a pair of objects. It is important to notice that features become truly distinctive features only after this transformation, while the features in the matrix $\mathbf{E}$ are not inherently common or distinctive. The weighted sum of these featurewise distances is the fitted distance for each pair of objects and is equal to $\hat{\mathbf{d}} = \mathbf{X}\hat{\boldsymbol{\eta}}$. Transforming the objects $\times$ feature matrix to the object pairs $\times$ features matrix is necessary to apply the multiple regression approach and, at the same time, provides a considerable reduction of the number of features to be generated, as will become clear later.

The multiple regression approach has been used earlier in the context of the common features model (Arabie & Carroll, 1980), and for tree models (Corter, 1996). However, the nonnegative least squares method has not been used by these authors, although they were aware of the problem. Only Arabie and Carroll (1980) address the problem by implementing a subroutine in the MAPCLUS algorithm that encourages the weights to become positive. These authors explain that the use of nonnegative least squares has been avoided explicitly because in the context of the iterative algorithm that is the basis of the MAPCLUS algorithm, it would reduce the number of clusters in the solution. We implemented the nonnegative least squares option in PROXGRAPH (the program used to fit FNM), not during the feature selection procedure, but for the situation where the features are supplied by the user. In that case, the use of nonnegative least squares has a considerable advantage because it opens the way to statistical inference by providing theoretical standard errors for the feature parameters (Frank & Heiser, in press a).

Table 4.2: Feature parameters ($\hat{\eta}$), standard errors, and 95% confidence intervals for Feature Network Model on *consonant* data with $R^2 = 0.61$

Features	$\hat{\eta}$	$\hat{\sigma}_\eta$	95% CI	
Constant	2.11	0.13	1.85	2.37
Voicing	1.22	0.11	1.01	1.43
Nasality	0.81	0.13	0.56	1.06
Affrication	0.12	0.11	-0.11	0.34
Duration	0.32	0.12	0.10	0.55
Place, middle	0.00	0.00	0.00	0.00
Place, front	0.10	0.07	-0.04	0.24
Place, back	0.26	0.10	0.06	0.45

Table 4.2 shows the feature discriminability parameters that result from minimizing the loss function in Equation 4.5, as well as the corresponding standard errors and 95% confidence intervals. The method to compute the standard errors and 95% *t*-intervals for inequality constrained feature parameters in the context of Feature Network Models has been described in (Frank & Heiser, in press a). The model with seven features has an $R^2 = 0.61$, and the values of the feature parameters lead to the conclusion that the most important categorizing criteria used by the participants were the following: *voicing*, *nasality*, *duration*, and *place, back*. The features *affrication*, *place, middle*, and *place, front* do not play an important role as follows from the 95% *t*-confidence intervals that show that the feature parameters of these features do not significantly differ from zero (see Table 4.2).

The feature distance parallels the path-length distance in a valued graph if one of the metric axioms, the triangle inequality, is reaching its limiting additive form $d_{ij} = d_{il} + d_{jl}$ when l is on the shortest path from i to j (Flament, 1963; Heiser, 1998). Hence, sorting out the additivities in the fitted feature distances and excluding edges that are sums of other edges results in a parsimonious subgraph of the complete graph. Figure 4.1 shows the Feature Network representation that results from the fitted distances on the *consonant* data. The phonemes are the vertices in the network and the estimated feature distances ($\hat{\mathbf{d}} = \mathbf{X}\hat{\boldsymbol{\eta}}$) are represented as additive counts of edge lengths in the graph, where the edge lengths are the feature parameters $\hat{\boldsymbol{\eta}}$. For display purposes the 7-dimensional feature network has been embedded in 3-dimensional Euclidean space using PROXSCAL[5] (a multidimensional scaling program distributed as part of the Categories package by SPSS, Meulman & Heiser, 1999). The solution of the common space was restricted by a linear combination of the feature variables, to be able to represent the features as vectors in the same space. The final network representation was obtained using the default options for 3-D plotting in Matlab. The three most important features (*voicing*, *nasality*, and *duration*) are represented as vectors in Figure 4.1, leading from the origin through the point with coordinates equal to the correlations of each feature with each of the three dimensions. The network clearly shows the importance of the *voicing* feature:

[5] with the interval transformation option and initialized with the simplex solution

all voiced consonants are on the left part of the network and well separated from the unvoiced consonants on the right part. The second important feature, *nasality*, separates the consonants *m* and *n* from the other consonants. The consonants *s*, ʃ, *z* and ʒ form a group with the shape of a rectangle and differ from the remaining 12 consonants because of the length of their pronunciation, described by the feature *duration*. The plus and minus signs on each vector designate the projection onto the vector of the centroids of the consonants that possess the feature (+) and the consonants that do not possess that feature (−).

Generating features with Gray codes

Given that features can be viewed as binary variables, a very straightforward way to produce all possible binary (0,1) features for *m* objects is to generate the binary codes for *m* bits of the integers 0 to $2^m - 1$, as illustrated in Table 4.3 for $m = 4$. Another, more restrictive way to produce the binary features, is to use the Gray code (Gray, 1953). A Gray code represents each number in the sequence of integers $\{0 \cdots 2^m - 1\}$

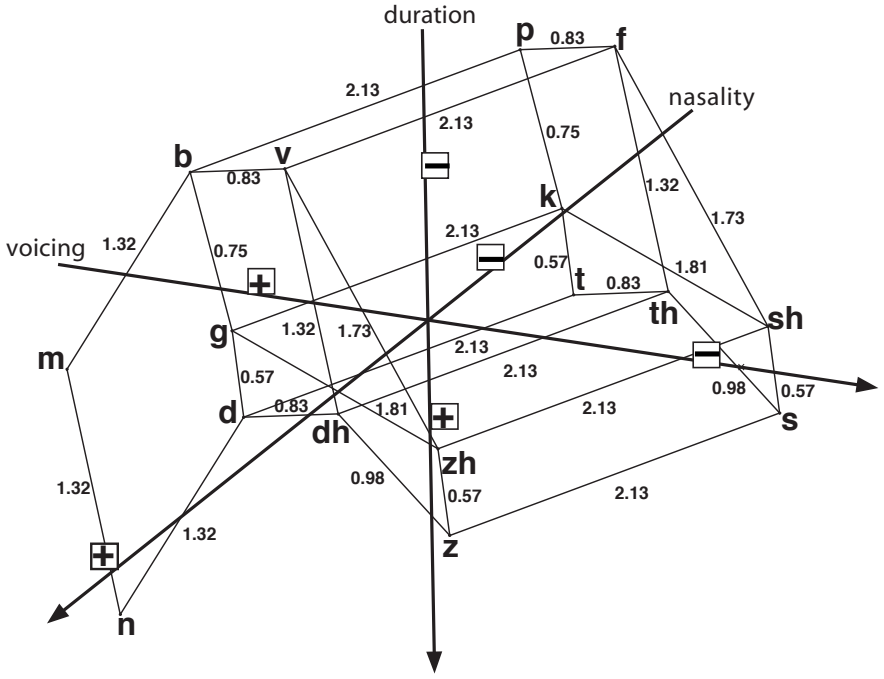

Figure 4.1: Feature Network representation for the *consonant* data with the three most important features (*voicing*, *nasality*, and *duration*) represented as vectors. The plus and minus signs designate the projections onto the vector of the centroids of the objects that possess the feature (+) and the objects that do not have that feature (-). (dh = ð; zh = ʒ; th = θ; sh = ʃ).

Table 4.3: Binary code and Gray code for 4 bits

Integer	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

as a binary vector of length m in an order such that adjacent integers have Gray code representations that differ only in one bit position, meaning that the transition from one integer to the next in the order requires changing just one bit at a time, which is called is called the *adjacency property* (*cf.* Gardner, 1972).

Table 4.3 shows the Gray coding corresponding to the integers based on 4 bits. There is a specific relationship between the Gray code and the binary code: Gray codes are binary codes arranged in a special order. Table 4.3 clearly shows the pattern of the flipping of one bit at a time, compared to the binary codes, where the transition from one integer to the next in the order is not restricted to the change of one bit. There are many ways to produce a binary sequence that has the adjacency property. The most common way to produce such a sequence is the so called *binary reflected Gray code* that starts with all bits zero and successively flips the right-most bit that produces a new string. Generating the binary reflected Gray codes works as follows. For m bits the list L_m starts with L_1 and produces the list $0, 1$. For $m > 1$ bits, L_m is formed by taking the first half of the list, L_{m-1} prepending a 0 to every number, then following that list by the reverse of L_{m-1} with a 1 prepended to every number (*cf.* Savage, 1997). For example, to obtain L_2, the list L_1 (0,1) is written forwards and backwards, producing 0, 1, 1, 0, and, prepending 0's to the first half and 1's to the second half, yields the L_2 list 00, 01, 11, 10.

In contrast to the more arbitrary binary codes, the Gray codes are directly related to the Feature Network Models. An m-bit Gray code is equal to a Hamiltonian cycle on an m-dimensional hypercube (Gilbert, 1958; Savage, 1997). It represents all the possible feature combinations for m objects. It is a cycle that visits each combination only once. The feature distance is a city-block metric on the binary coordinates of this same space. Since adjacent Gray codes differ by only one bit, feature distances

of three consecutive Gray codes are additive. The binary coordinates represent the feature pattern of each object. Based on their feature pattern, the objects have their place in this m-dimensional hypercube.

The Gray code shares with the binary code the nice property that the second half of the list of the codes is the complement of the first half. Table 4.3 shows this property: the pattern of the Gray codes (and the binary code) representing the integers 8 to 15 is the negative of the pattern of the integers 0 to 7. The fact that the second half of the Gray codes for m objects is the complement of the first half constitutes a useful property in the context of the Feature Network Models. Since complementary features yield the same **X**-matrix and consequently the same $\hat{\eta}$ values as the original features, only half of the number of 2^m features needs to be generated. This property holds for the distinctive features only, and not for the common features, where it would be necessary to generate the complete Gray code. Further reductions can be obtained by discarding the feature with zeros only, because it has no meaning in the Feature Network Models. The feature with ones only (the universal feature) is always located in the second half of the Gray code and will therefore not be part of the set of generated features. However, PROXGRAPH, the program used to fit FNM (programmed in Matlab), has the option of adding the universal feature to the model. A remark has to be made about a special category of features, the unique features, which describe only one object (having a 1 for that particular object and zero values for the remaining objects). In the common features model the presence of one or more unique features in the object $\times$ features matrix **E** leads to a zero feature product in the predictor set and is one of the problems to be avoided in, for example, the MAPCLUS algorithm (Arabie & Carroll, 1980). The FNM that use featurewise distances does not have this inconvenience and therefore all Gray codes representing the unique features can be part of the complete feature set.

Summarizing, complete enumeration of the distinctive features for m objects amounts to generating $\frac{1}{2}(2^m) - 1$ Gray codes, using the integers $\{1 \cdots \frac{1}{2}(2^m) - 1\}$, and forming the complete set of featurewise distances $\mathcal{D}$ that contains a total number of $T_{\mathcal{D}} = \frac{1}{2}(2^m) - 1$ predictors. Taking the set $\mathcal{D}$ as the starting point of the feature subset selection process constitutes a considerably smaller problem than would be the case for the generation of predictors in a univariate multiple regression problem with arbitrary binary predictors. In that case the number of predictors to be enumerated amounts to 2^n, where n is equal to $\frac{1}{2}m(m-1)$, which shows the proportion of additional predictors that are needed. The possibility of using the transformation from the matrix **E** to the matrix **X** allows for this reduction of the number of predictors to be generated.

However, there are limitations to the total number of distinctive features that can be handled because the set $T_{\mathcal{D}}$ of predictors grows considerably with increasing number of objects m. For example, for $m = 20$ the set $T_{\mathcal{D}}$ contains $\frac{1}{2}(2^m) - 1 = \frac{1}{2}(2^{20}) - 1$ or about a half million distinctive features, growing to about 1 million for $m = 21$, becoming more than 2 million for $m = 22$ and exceeding 4 million for $m = 23$. A set of $\frac{1}{2}(2^{22}) - 1$ (= about 2 million) predictors is the maximum number that the current implementation of the predictor selection algorithm, the Positive Lasso, can handle simultaneously.

To generate the Gray codes within PROXGRAPH, we used a Matlab transcription by Burkardt of the original algorithms for generating Gray codes in Nijenhuis and Wilf (1978) (see http://www.csit.fsu.edu/ burkardt/, Fortran and C++ files of the same algorithms are also available at this site). Both binary code and Gray code have the convenient attribute that features can be (re)produced by a simple integer or rank number. This property saves computer memory because it is not necessary to save the entire sequence of $\frac{1}{2}(2^m) - 1$ features, since the original feature set can be retrieved by simply keeping track of the corresponding integer or rank number. Another advantage of saving the integer or rank numbers is the possibility of getting back the original features after transformations to featurewise distances have been applied on those features. In the Feature Network Models one important transformation performed on the features is the transformation from the $(m \times T)$ matrix $\mathbf{E}$ representing the T features that describe the m objects to the matrix $\mathbf{X}$ of size $(n \times T)$, that contains the symmetric set difference for each of the $n = \frac{1}{2}m(m-1)$ object pairs (Equation 4.6). This matrix $\mathbf{X}$ is also the format for the features when submitted to the feature selection algorithm. The problem with this transformation is that it is not reversible because the results are not unique. In the simple example of one feature and two objects the result 0 can come from $\mathbf{x}_{12} = |1-1|$, where both objects have the feature, or from $\mathbf{x}_{12} = |0-0|$, where neither of the objects possesses the feature. The result 1 is not unique either. It means that one of the two objects has the feature, but it is not clear which object has the feature. Therefore, saving the rank numbers of the features in the set before applying the transformation, makes it possible to reproduce the original feature matrix at the end of the entire feature selection process.

Selecting a subset of features with the Positive Lasso

Above we have shown that the FNM can be considered as a univariate multiple linear regression problem with positivity constraints on the feature discriminability parameters (see Equations 4.3, 4.4, and 4.5). When the features are known in advance and their number is reasonably small, the feature discriminability parameters can be obtained directly by minimizing the nonnegative least squares loss function of Equation 4.5. However, in the case of unknown features, the Gray codes are used to generate a very large number of features, and, as a result, the simple nonnegative least squares loss function cannot be used. The large number of features calls for a variable selection method. There are many methods available for variable selection, see for example a recent review by Guyon and Elisseeff (2003). Given the multiple regression context of the FNM, we have chosen the least absolute shrinkage and selection operator (Lasso). The Lasso is a constrained version of ordinary least squares (OLS) and minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant (Tibshirani, 1996). Let $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$ be n-vectors representing the T featurewise distances with n equal to the number of unique pairs of objects, and $\boldsymbol{\delta}$ the n vector of dissimilarities. It is assumed that the featurewise distances have been standardized to have mean 0 and

unit length and that the response variable ($\boldsymbol{\delta}$) has mean 0:

$$\sum_{l=1}^{n} \delta_l = 0, \quad \sum_{l=1}^{n} x_{nt} = 0 \quad \text{and} \quad \sum_{l=1}^{n} x_{lt}^2 = 1 \quad \text{for} \quad t = 1, 2, \cdots, T. \tag{4.7}$$

Applied to the context of the FNM, the Lasso loss function can be written in the following way:

$$\min_{\boldsymbol{\eta}_{\text{L}}} = \|\boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta}_{\text{L}}\|^2 \qquad \text{subject to} \quad G(\boldsymbol{\eta}_{\text{L}}) \leqslant b, \tag{4.8}$$

where the constraint $G(\boldsymbol{\eta}_{\text{L}}) = \sum_{t=1}^{T} |\eta_t|$ and $b \geq 0$ is the tuning parameter that controls the amount of shrinkage. If $\hat{\boldsymbol{\eta}}^0$ is the vector of ordinary least squares estimates and $b_0 = \sum_{t=1}^{T} |\hat{\eta}_t^0|$, the Lasso estimates become the ordinary least squares estimates for values of $b > b_0$. On the other hand, values of $b < b_0$ will cause shrinkage of the solutions toward 0, and some of the coefficients will become exactly equal to 0. This effect constitutes the parsimony property that characterizes the Lasso compared to ridge regression, which is related to the Lasso and is probably more generally known.

For any given constraint value b in the path of Lasso solutions[6], only a subset of the features has non-zero values of the regression coefficients $\hat{\eta}$. While ridge regression also shrinks coefficients, it does not, however, set any coefficients to 0 and, as a result, does not lead to more simple models. The differences in the nature of shrinkage between the Lasso and ridge regression result from the constraints used in both methods. Both methods use the residual sum of squares loss function, but where the Lasso uses the constraint $\sum_{t=1}^{T} |\eta_t|$, ridge regression uses $\sum_{t=1}^{T} \eta_t^2$ instead (see for more details: Hastie et al., 2001; Tibshirani, 1996). From the viewpoint of geometry, the Lasso constraint leads to a constraint region with corners and flat edges, while ridge regression leads to round shaped constraint regions, see Figure 4.2. The residual sum of squares function has elliptical contours and both methods find the first point where these elliptical contours hit the constraint region. In the case of the Lasso, when the elliptical contours hit a corner, some of the estimated parameters become exactly 0, while in the case of ridge regression the estimated parameters will never become 0 because the elliptical contours will never hit a corner.

In general, shrinkage improves prediction accuracy, trading off decreased variance for increased bias, Hastie et al. (2001). For the special case of the Lasso, shrinkage leads to more parsimonous models because some coefficients become exactly zero. Another advantage of the Lasso, especially useful for the FNM context, is that it does not suffer from overfit or highly correlated settings because it avoids the explicit use of the OLS estimates. This means that the design matrix $\mathbf{X}$ need not be of full rank, which is very convenient in a situation with a very large number of featurewise distances.

[6] In contrast to ordinary least squares, the Lasso does not yield a single solution but a path of solutions depending on the values of the tuning parameter b. Typically, Lasso solutions are computed for several values of b, ranging from $b = 0$ to $b = b_0$. An example of a path of Lasso solutions can be viewed in Figure 4.3, starting with $b = 0$, which forces all coefficients to become zero, and ending with the value of $b = b_0$ equal to the sum of the coefficients of the ordinary least squares solution.

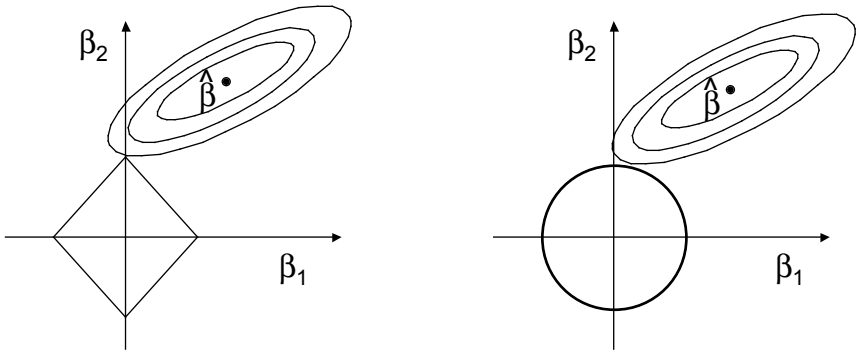

Figure 4.2: Graphs of estimation for the Lasso (left) and ridge regression (right) with contours of the least squares error functions (the ellipses) and the constraint regions, the diamond for the Lasso and the disk for ridge regression. The corresponding constraint functions are equal to $|\beta_1| + |\beta_2| \leqslant b$ for the Lasso and $\beta_1^2 + \beta_2^2 \leqslant b^2$ for ridge regression . It is clear that only the constraint function of the Lasso can force the $\hat{\beta}$-values to become exactly equal to 0. (The graphs are adapted from Hastie et al. (2001), p. 71).

The computation of Lasso solutions is a quadratic programming problem, and can be solved by numerical analysis algorithms, but the LARS or Least Angle Regression (Efron et al., 2004) is a better approach. The LARS algorithm works as follows. It starts with all feature parameters of the vector $\hat{\boldsymbol{\eta}} = (\hat{\eta}_1, \hat{\eta}_2, \cdots, \hat{\eta}_T)'$ equal to 0. The next step is to find the predictor $\mathbf{x}_t$ most correlated with response $\boldsymbol{\delta}$, which is added into the model. The residuals $\mathbf{r} = \hat{\mathbf{d}} - \boldsymbol{\delta}$ are calculated and the parameter $\hat{\eta}_t$ is increased in the direction of the sign of its correlation with $\boldsymbol{\delta}$ until some other feature $\mathbf{x}_k$ has as much correlation with the current residual vector as does $\mathbf{x}_t$. The feature parameters $(\hat{\eta}_t, \hat{\eta}_k)$ are increased in their joint least squares direction, until some other feature $\mathbf{x}_q$ has as much correlation with the current residual. The just described steps are repeated until all features have been entered in the model and the process stops when $\text{corr}(\mathbf{r}, \mathbf{x}_t) = 0 \; \forall \; t$, which corresponds to the OLS solution. In the situation where the number of predictors T exceeds the number of observations n, the LARS algorithm terminates at the saturated least squares fit after $n - 1$ predictors have entered the active set (see, for more details, Efron et al., 2004, p. 444). The number $n - 1$ follows from mean centering the columns of the matrix of predictors $\mathbf{X}$ which results in a row-rank equal to $n - 1$. It should be noted however, that although the model contains no more than $n - 1$ predictors, the number of *different* predictors that have entered the model during the complete sequence of solutions is typically greater than $n - 1$.

LARS provides an efficient way to compute the Lasso sequence of solutions simultaneously for all values of b, as b varies from 0 to infinity by applying the following modification: if a non-zero parameter becomes zero, it is removed from the active set of features and the joint direction is recomputed. The implementation

of LARS in *R* also allows for the transformation of the Lasso into a Positive Lasso necessary for the FNM where all feature discriminability parameters should be positive. We applied the procedure described in Efron et al. (2004, section 3.4, p. 421) to the LARS algorithm programmed in *R*. The result is the solution of the following minimization function:

$$\min_{\boldsymbol{\eta}_{\text{PL}}} = \|\boldsymbol{\delta} - \mathbf{X}\boldsymbol{\eta}_{\text{PL}}\|^2 \quad \text{subject to} \quad G(\boldsymbol{\eta}_{\text{PL}}) \leqslant b \quad \text{and all} \quad \eta_t \geqslant 0, \tag{4.9}$$

where the constraint in Equation 4.8 is extended with a positivity constraint for the feature discriminability parameters.

Selecting the number of features with an AIC criterion

The Lasso and the Positive Lasso do not yield a single solution $\hat{\boldsymbol{\eta}}$, but a path of possible solutions defined by the continuum depending on the values of the tuning parameter b, which represents the amount of shrinkage. Choosing a value for b leads automatically to the choice of the number of features in the model, i.e. the number of features with nonzero $\hat{\eta}$-values. The problem is to choose a good value for the a priori unknown b, such that the corresponding model minimizes the prediction error.

Efron et al. (2004) proved that for some LARS estimators, the best value for b can be found with an adaptation of Mallows' C_p statistic (Mallows, 1973, 1995), where the step number of the LARS algorithm is used as an estimate for the degrees of freedom of the corresponding model. For the Lasso and the Positive Lasso estimators, the step number of the LARS algorithm cannot be used as an estimate for the degrees of freedom because the total number of steps can exceed the total number of predictors in the full model. However, recently, Zou, Hastie, and Tibshirani (2006) showed that the number of non-zero coefficients is an unbiased estimate for the degrees of freedom for the Lasso, an informative measurement of model complexity, with no special assumptions on the predictors. This estimate for the degrees of freedom in the Lasso can be used to estimate the prediction error of each of the models along the path of Lasso solutions by the following AIC criterion, derived especially for the Lasso (Zou et al., 2006):

$$AIC_{\text{L}} = \frac{\|\boldsymbol{\delta} - \hat{\mathbf{d}}\|^2}{n} + \frac{2}{n}\widehat{df}(\hat{\mathbf{d}})\sigma^2_{\text{L}}, \tag{4.10}$$

where $\hat{\mathbf{d}} = \mathbf{X}\hat{\boldsymbol{\eta}}_{\text{L}}$, and the error variance σ^2_{L}, if unknown, is replaced with an estimate based on the largest model. In the case where the number of predictors exceeds the number of observations, the largest model in the total sequence of Lasso solutions resulting from the LARS algorithm, involves at maximum $n-1$ predictors. Since the largest model is a (nearly) saturated model, the error variance is very close to zero. Therefore, we estimated σ^2_{L} in Equation 4.10 by taking the mean of the error variances of all models in the sequence of Lasso solutions.

The AIC_{L} criterion, which approximates the Mallows' C_p statistic (Mallows, 1973, 1995) closely, has been shown to offer substantially better accuracy than cross-validation and related nonparametric methods, if one is willing to assume the model

is correct (Efron et al., 2004; Zou et al., 2006). We used the AIC_{L} criterion to select the best model for the Positive Lasso solutions and to assess the prediction error, assuming that the theory about the effective number of non-zero parameters applies for the Positive Lasso as well. To our knowledge it is the best criterion available at the moment.

Figure 4.3 shows the results of the modifications of the Lasso-LARS algorithm into the Positive Lasso as in Equation 4.9 using the theoretical (phonetic) features of the *consonant* data (Table 4.1). The left top panel shows the paths of the Lasso estimates of the feature discriminability parameters against the degrees of freedom expressing the effective number of nonzero parameters. Feature 5 (*place, middle*) obtains negative feature discriminability parameters along the path. The right top

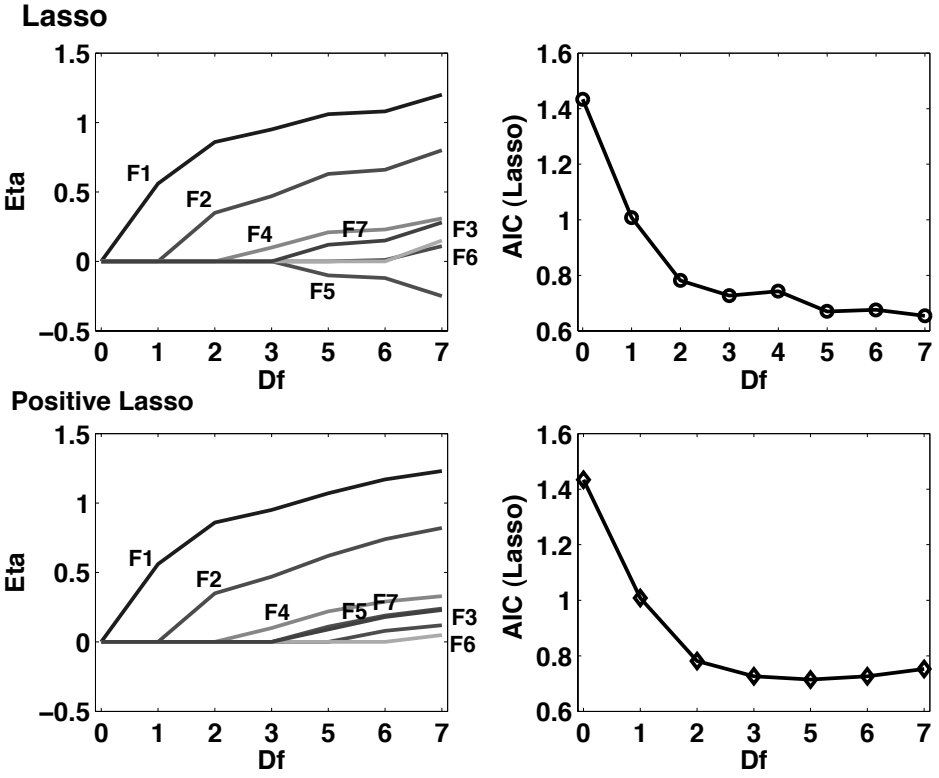

Figure 4.3: Estimates of feature parameters for the *consonant data*. *Top panels*: trajectories of the Lasso estimates $\hat{\eta}_{\mathrm{L}}$ (left panel) and the AIC_{L} values plotted against the effective number of parameters (= df) of the Lasso algorithm (right panel). The model with lowest AIC_{L} value (= 0.65) contains all 7 features. *Lower panels*: trajectories of the Positive Lasso estimates $\hat{\eta}_{\mathrm{PL}}$ (left panel) and the adjusted AIC_{L} values plotted against the effective number of parameters (= df) of the Positive Lasso algorithm (right panel). The model with lowest AIC_{L} value (= 0.71) has 5 features.

Table 4.4: Estimates of feature discriminability parameters ($\hat{\eta}_{\text{ICLS}}$ = ICLS, $\hat{\eta}_{\text{L}}$ = Lasso, and $\hat{\eta}_{\text{PL}}$ = Positive Lasso) for the *consonant data*

Features	$\hat{\eta}_{\text{ICLS}}$	$\hat{\eta}_{\text{L}}$	$\hat{\eta}_{\text{PL}}$
intercept	2.11	2.22	2.39
Voicing	1.22	1.20	1.07
Nasality	0.81	0.80	0.62
Affrication	0.12	0.11	0.00
Duration	0.32	0.31	0.22
Place, middle	0.00	-0.25	0.11
Place, front	0.10	0.15	0.00
Place, back	0.26	0.28	0.09

panel shows the AIC_{L} values at each step of the iterations, plotted against the estimated degrees of freedom, represented by the effective number of non-zero parameters as in Equation 4.10. The AIC_{L} curve, which also represents the estimates of prediction error, shows that the best model occurs at 7 df, the model that contains all features. The complete model corresponds to the ordinary least squares solution because the Lasso always converges to it. The lower left panel shows the path of the Positive Lasso estimates of the feature discriminability parameters and it is clear that all trajectories stay in the positive part of the parameter space. The AIC_{L} curve in the lower right panel indicates that the best model occurs at $df = 5$ with only 5 of the 7 features present in the model.

Table 4.4 displays the estimates of feature discriminability parameters according to the best Lasso model and the best Positive Lasso model based on the AIC_{L} curves compared to the inequality constrained least squares estimates obtained with Equation 4.5. In this case the Lasso estimates $\hat{\eta}_{\text{L}}$ are equal to the ordinary least squares estimates, without positivity constraints, and yield a negative coefficient value for feature *place, middle*. The Positive Lasso estimates $\hat{\eta}_{\text{PL}}$ show that the two features *affrication* and *place, front* have coefficient values equal to zero as a result of activated positivity constraints, and consequently, these two features are not part of the model. This finding confirms the results of the 95% CI presented before (Table 4.2) showing that the feature parameters ($\hat{\eta}_{\text{ICLS}}$) of these two features do not significantly differ from zero[7].

[7]It should be noted that from the perspective of phonetic theory, it is rather unusual that the feature *affrication* disappears from the model and should probably be ascribed to the experimental conditions used by Miller and Nicely (1955). The authors presented the consonants under 6 different signal-to-noise conditions and, as a result, the non-fricative consonants become contaminated with noise and are no longer distinguishable from the fricatives. The same experimental conditions could also explain the fact that *voicing* has so much influence, while it is known as a phonetic feature that is easily lost during perception.

Generating features by taking a random sample combined with a filter

When the number of objects m exceeds 22, it is not possible to generate the complete set of distinctive features. In that case, we propose to take a sample from the total number of rank numbers representing the Gray codes associated with the number of objects. This sample might still be too large to be submitted to the Positive Lasso algorithm and necessitates some preselection strategy. Preselection is often used as a preprocessing step before variable subset selection. A review on this topic has been given by Guyon and Elisseeff (2003). A common preselection strategy is the ranking method that performs this preprocessing step by selecting variables that have high values on a scoring function, usually the coefficient of determination or R^2 (Guyon & Elisseeff, 2003). The variables are sorted in decreasing order based on their values on the scoring function. To build a predictor, nested subsets are constructed which incorporate progressively more variables of decreasing relevance. Other scoring functions are the correlation, where positively correlated variables are top ranked and negatively correlated variables bottom ranked. We propose the use of the regression coefficient, or, in the context of FNM, discriminability parameter $\hat{\eta}$, which is a scaled version of the correlation coefficient as can be seen in the following relation (*cf.* Draper & Smith, 1998, p. 42):

$$\hat{\eta} = \frac{s_{\boldsymbol{\delta}}}{s_{\mathbf{x}}} r_{\boldsymbol{\delta}\mathbf{x}}, \tag{4.11}$$

where $s_{\boldsymbol{\delta}}$ and $s_{\mathbf{x}}$ are the standard deviations of the dependent variable $\boldsymbol{\delta}$ and the predictor variable $\mathbf{x}$, and $r_{\boldsymbol{\delta}\mathbf{x}}$ is the correlation between the dependent variable and the predictor variable. It is clear that when both $\boldsymbol{\delta}$ and $\mathbf{x}$ are standardized, as required for the Positive Lasso, the regression coefficient is equal to the correlation.

Example of feature generation and selection on the *consonant* data

The previous section showed the results of the Positive Lasso on the a priori phonetic features of the *consonant* data. In many data analytic situations, the features are not given by theory. This section shows an example of feature generation using Gray codes followed by feature selection with the Positive Lasso on the same *consonant* data. First, all possible distinctive features were generated with the number of Gray codes equal to $\frac{1}{2}(2^{16}) - 1 = 32,767$ because there are 16 consonants, yielding a $16 \times 32,767$ matrix of objects by features. After transformation of this matrix into the $120 \times 32,767$ matrix $\mathbf{X}$ using Equation 4.6, the complete set of featurewise distances was analyzed with the Positive Lasso algorithm.

Figure 4.4 shows that the AIC_{L} curve attains its lowest value ($= 0.51$) at the model with 7 features. Table 4.5 shows the values of the feature discriminability parameters for the feature matrix obtained from phonetic theory and for the feature matrix resulting from the Positive Lasso algorithm. The model resulting from the Positive Lasso has higher fit ($R^2 = 0.70$) and lower prediction error values compared to the model based on phonetic theory.

Table 4.6 displays the features of the model selected by the Positive Lasso algorithm juxtaposed to the 7 features based on phonetic theory. Comparing the features

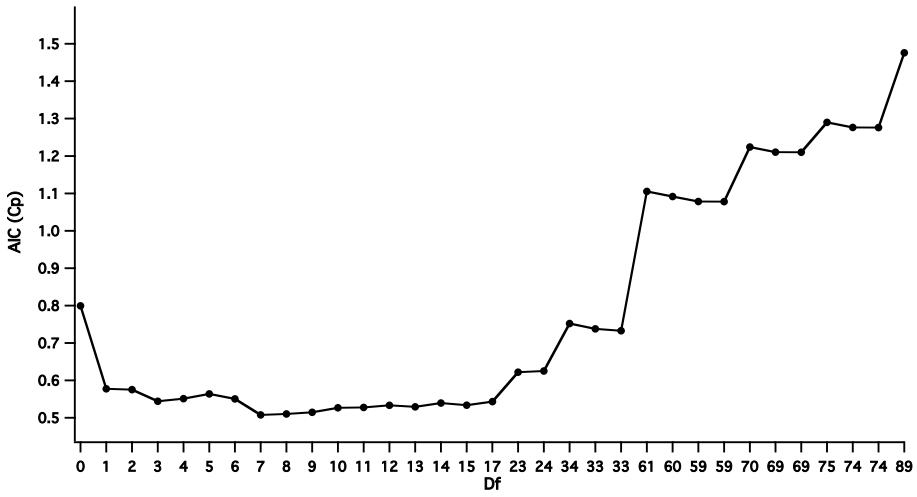

Figure 4.4: AIC_{L}-plot for the *consonant* data using all possible features generated with Gray codes ($T = 32,767$). The lowest AIC_{L} value ($= 0.51$) points to a model with 7 features.

from phonetic theory to the features resulting from the Positive Lasso, leads to the conclusion that the two feature sets are very different from each other, except for the first feature that represent the phonetic property *voicing*. The remaining features selected by the Positive Lasso do not seem to be related to the theoretic phonetic properties of the consonants. However, the network representation of the feature set selected by the Positive Lasso displayed in Figure 4.5 does make sense in terms of phonetics: there is a clear distinction between the voiced consonants (*m*, *n*, *b*, *d*, ð, *g*, *v*, *z*, ʒ) on the left part of the configuration and the unvoiced consonants on the right part. The nasals (*m*, *n*) form a distinct cluster, showing the importance of the phonetic property *nasality*. The cluster (*s*, *θ*) represents middle voiceless consonants and the cluster (ð, *b*, *v*) front and middle voiced consonants. Another cluster that can be distinguished comprise the voiceless plosives, (*p*, *t*, *k*) opposed to the three voiced plosives (*b*, *d*, *g*)[8]. The clusters just described correspond to clusters found with ADCLUS by Shepard and Arabie (1979) and with MAPCLUS by Arabie and Carroll (1980).

[8]From the perspective from phonetics, the same remarks that were made in footnote[7] for the solution of the set of 7 theoretic features in Figure 4.3 and Table 4.4 also apply to this solution.

4.3 Simulation study

In this section we report a Monte Carlo experiment that evaluated the performance of our method given that the true feature structure (the true model) is known in advance. The first question that will be addressed is the following: does the Positive Lasso select the correct subset of features given that the true feature set is known, under different data analytic conditions such as error, n/T ratio (number of object pairs compared to the number of features), and the size of the feature discriminability parameters? The performance criterion of this study is the proportion of recovery of the true features, measured by Gray code rank number. We verified the baseline condition of the proportion of recovery on error-free data, which resulted in complete recovery of the correct features for the experimental conditions. Another question addressed by the simulation study is: how does the method of random sampling from Gray codes combined with a filter perform compared to the complete enumeration method? Proportion of recovery of true features is not a useful performance criterion in this situation because random samples of features are taken and that would merely result in testing the performance of the pseudo-random number generator, instead of testing the performance of the method. Instead, the following measures serve as outcome: the effective number of parameters (Df) and the prediction error (AIC_{L}).

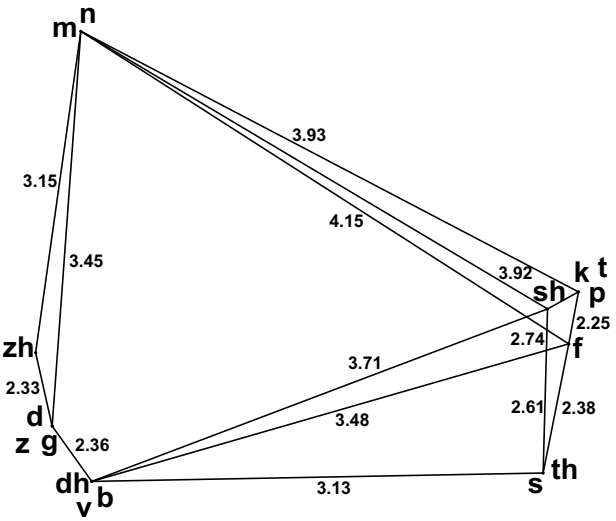

Figure 4.5: Feature Network representation for the *consonant* data based on the feature matrix selected by the Positive Lasso displayed in Table 4.6. (dh = ð; zh = ʒ; th = θ; sh = ʃ).

Table 4.5: Positive Lasso estimates, R^2, and prediction error (K-fold cross-validation) for the features from phonetic theory (left) and for the features selected from the complete set of distinctive features (right)

Features from phonetic theory		*Features selected from the complete set*	
Features	$\hat{\eta}_{\text{PL}}$	Features	$\hat{\eta}_{\text{PL}}$
intercept	2.39	*intercept*	2.03
Voicing	1.07	F1	0.91
Nasality	0.62	F2	0.33
Affrication	0.00	F3	0.30
Duration	0.22	F4	0.35
Place, middle	0.24	F5	0.22
Place, front	0.00	F6	0.36
Place, back	0.09	F7	0.19
Model fit			
R^2	0.56		0.70
Prediction error (standard error) based on K-fold cross-validation			
K=5	0.35 (0.08)		0.29 (0.04)
K=10	0.35 (0.07)		0.27 (0.05)

Method for simulation study

The experimental conditions of this simulation study result from the cross classification of five experimental variables with two levels each. For each experimental condition, a total of 50 simulation samples were generated. The first experimental variable is the *number of objects* m = 12 or 24. The second experimental variable is the ratio of the number of observations n and the number of features T and has two levels: $n/T = 16$ and $n/T = 8$. Given the two levels of *number of objects*, the number of observations n is equal to the number of object pairs $n = \frac{1}{2}m(m-1)$. For the 12 objects condition, which has $n = 66$, the number of features needed to obtain the two n/T ratios is 4 and 8. For the 24 objects condition (with n equal to 276) the number of features needed is equal to 17 and 35. The third experimental variable is the *size of the feature discriminability parameters* with two levels: medium values (M) and a combination of small and large values (S + L). Depending on the number of features needed the following patterns of 4 feature discriminability parameters is repeated. For the medium values conditions the pattern is $\{2.0, 2.5, 1.5, 3.0\}$ and for the small + large values the pattern is $\{6.0, 0.2, 0.5, 0.3\}$. The fourth experimental variable is the *amount of error* added to the data, and comes in two levels: 0.05 (low) and 0.35 (high). The fifth experimental variable is the *feature generation strategy*, which consists of either generating the whole set of possible features using half of the complete Gray code sequence, or a set of the 100 best features based on the filter criterion of largest separate $\hat{\eta}$ value selected from a large random sample (30%) of all possible features.

Using the whole set of possible features is only feasible for the 12 objects condition. As a result, the random sample strategy was applied to all 16 combinations of the experimental variables (*number of objects*, n/T ratio, *size of the feature discriminability parameters*, *amount of error*), and the complete set of features strategy was used for the 12 objects condition only, leading to 6 experimental conditions obtained from the combination of n/T ratio, *size of the feature discriminability parameters* and *amount of error*. The 50 simulation samples in each experimental condition were submitted to the Positive Lasso algorithm and the performance of the selected models was evaluated by the following measures that serve as outcome variables: the effective number of features (Df = the number of nonzero parameters) and the prediction error AIC_L obtained with Equation 4.10.

Generating true configurations and sampling dissimilarities

The true configurations are defined by the three experimental variables *number of objects*, n/T ratio, and *size of the feature discriminability parameters*. The combination of the levels of these three experimental variables results into 8 experimental conditions. For each of these conditions, a feature matrix was created by taking a random sample from the total number of feature patterns for a given number of features (4, 8, 17, and 35). This method generates features for the number of bits equal to the number of features and results in a matrix of size $[\frac{1}{2}(2^T) - 1] \times T$ that represents the

Table 4.6: Matrices of features based on phonetic theory (left) and of features selected by the Positive Lasso (right)

	Features from phonetic theory							*Features selected by Positive Lasso*						
Consonants	F1	F2	F3	F4	F5	F6	F7	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*	*F7*
p	0	0	0	0	0	1	0	1	1	1	0	0	1	0
t	0	0	0	0	1	0	0	1	1	1	0	0	1	0
k	0	0	0	0	0	0	1	1	1	1	0	0	1	0
f	0	0	1	0	0	1	0	1	1	1	0	1	1	0
θ	0	0	1	0	1	0	0	1	1	1	1	1	1	0
s	0	0	1	1	1	0	0	1	1	1	1	1	1	0
ʃ	0	0	1	1	0	0	1	1	1	1	1	0	0	0
b	1	0	0	0	0	1	0	0	1	1	1	1	1	1
d	1	0	0	0	1	0	0	0	0	1	1	1	1	1
g	1	0	0	0	0	0	1	0	0	1	1	1	1	1
v	1	0	1	0	0	1	0	0	1	1	1	1	1	1
ð	1	0	1	0	1	0	0	0	1	1	1	1	1	1
z	1	0	1	1	1	0	0	0	0	1	1	1	1	1
ʒ	1	0	1	1	0	0	1	0	0	0	1	1	1	1
m	1	1	0	0	0	1	0	0	0	0	0	0	0	0
n	1	1	0	0	1	0	0	0	0	0	0	0	0	0

Table 4.7: Feature matrices for 12 objects and rank numbers used to construct the true configurations for the simulation study

4 features condition				*8 features condition*							
F1	F2	F3	F4	F1	F2	F3	F4	F5	F6	F7	F8
0	1	0	1	1	1	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	1
0	0	1	1	0	1	0	0	0	1	0	1
0	1	1	1	1	0	0	1	0	0	1	0
1	1	1	1	1	1	1	1	0	0	0	0
1	1	1	0	1	0	0	0	0	0	1	0
0	1	1	0	1	1	0	1	1	1	1	1
1	0	1	0	1	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0
1161	322	688	86	691	415	1921	444	1533	1568	1729	495

complete lattice of possible feature patterns for the given number of features. Selecting at random 12 or 24 (corresponding to the number of objects) feature patterns yields a connected network. It should be noted that this method is slightly different from the way the Gray codes are used to create the complete set of distinctive features, where Gray codes are generated for the number of bits equal to the number of *objects* instead of the number of *features*. Table 4.7 shows the resulting feature matrices for 12 objects and 4 or 8 features with on the bottom row the corresponding Gray code rank numbers. The distances of the true configurations were computed using the levels of the experimental variable *size of the feature discriminability parameters*. The network representations for the 12 objects with 4 and 8 features and the two different levels of the sizes of feature discriminability parameters are displayed in Figure 4.6.

The true distances $\mathbf{d}$ for these configurations were obtained with $\mathbf{d} = \mathbf{X}\boldsymbol{\eta}$, where $\boldsymbol{\eta}$ represents the experimental values of the feature discriminability parameters, and $\mathbf{X}$ results from the transformation from Equation 4.6 applied on the feature matrices displayed in Table 4.7. The feature matrices and configurations for the 24 objects condition were obtained in exactly the same way. For each of the experimental conditions 50 samples of dissimilarities were obtained with the two levels of error using the binomial distribution to ensure positive dissimilarity values that follow a normal distribution. The details of the method of sampling from the binomial distribution are described in Frank and Heiser (in press a).

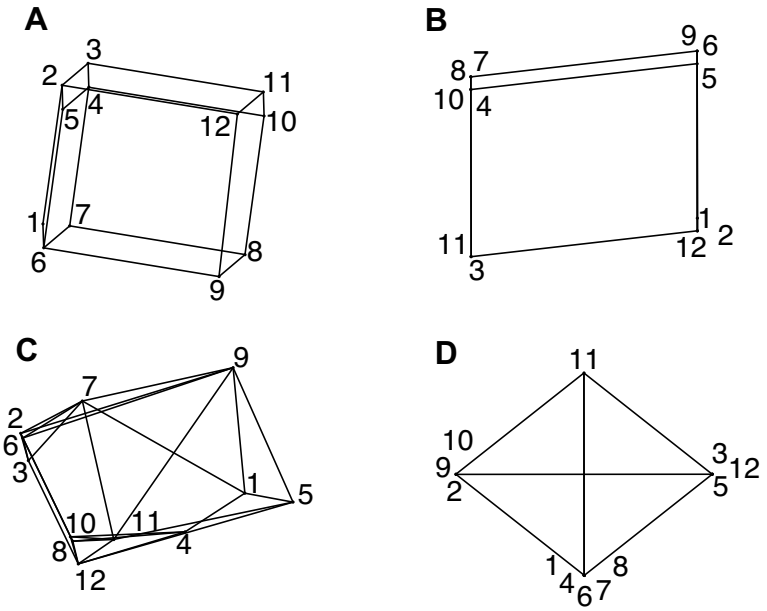

Figure 4.6: Feature network plots for the experimental conditions for 12 objects. A = 4 features, medium $\boldsymbol{\eta}$; B = 4 features, small + large $\boldsymbol{\eta}$; C = 8 features, medium $\boldsymbol{\eta}$; D = 8 features, small + large $\boldsymbol{\eta}$.

Results simulation study

Simulation results of the strategy using the complete set of Gray codes

This section answers the question whether the Positive Lasso selects the correct subset of features given that the true feature set is known, under different data analytic conditions. Table 4.8 shows the proportions of correctly recovered true feature rank numbers for the 50 simulation samples under the experimental conditions defined by combined levels of error, number of features and feature parameter size. In general, the true features are better recovered in the medium feature parameter values condition, compared to the combination of small and large feature parameter values.

When the feature parameter values all have medium size, the recovery is mainly affected by the ratio of the number of features compared to the number of observations (n/T). When the true number of features is small (n/T=16), there is perfect recovery of the features, regardless of the error level. In the condition of larger number of features compared to the number of observations (n/T=8) the true number of features is less well recovered. In the low error condition the proportions range from 0.86 to 1.00, with perfect recovery for the features with the highest true feature parameter values. In the high error condition, the features with the highest true fea-

ture parameter values are perfectly recovered, while the features with the lower true feature parameter values are less well recovered with proportions ranging from 0.00 to 0.62. The condition of combined small and large feature parameter values shows a different pattern: the features with large feature parameter values are perfectly recovered in all conditions formed by the combination of error level and the ratio of number of features compared to the number of observations. The features associated with small feature parameter values are recovered with small proportions in the condition of small number of features compared to the number of observations (n/T=16), and are almost never recovered in the condition of larger number of features compared to the number of observations (n/T=8).

Additional information on fit and effective number of features in the selected models is displayed in Figure 4.7, which shows the distributions of 50 simulation samples on 12 objects for all experimental conditions, using the complete set of distinctive features. The panels on the first row represent the effective number of features (= *Df*) selected by the Positive Lasso for each simulation sample and the true

Table 4.8: Proportion of correctly recovered features from the complete set of distinctive features under combined levels of error (L = low; H = high), the ratio of the number of object pairs and the number of features (= n/T ratio), and feature parameter (η) sizes, medium and small + large.

$n/T = 16$		*medium η values*							
		2.0	2.5	1.5	3.0				
	Error								
	L	1.00	1.00	1.00	1.00				
	H	1.00	1.00	1.00	1.00				
		small + large η values							
		6.0	0.2	0.5	0.3				
	Error								
	L	1.00	0.02	1.00	0.36				
	H	1.00	0.00	0.22	0.00				
$n/T = 8$		*medium η values*							
		2.0	2.5	1.5	3.0	2.0	2.5	1.5	3.0
	Error								
	L	0.98	1.00	0.94	1.00	0.94	1.00	0.86	1.00
	H	0.62	1.00	0.16	1.00	0.42	1.00	0.00	1.00
		small + large η values							
		6.0	0.2	0.5	0.3	6.0	0.2	0.5	0.3
	Error								
	L	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	H	1.00	0.00	0.00	0.00	1.00	0.02	0.00	0.00

number of features is represented as a dashed line. The panels on the second row show the associated AIC_L values, which are measures of prediction error for the selected models. Each of the eight panels represents the two error levels, low (L) and high (H). The four panels on the left correspond to the condition with medium η values and the four panels on the right (first row and second row) correspond to the condition with small + large η values.

Since the pattern of the outcomes differs in these two levels of η values, we describe the results separately, beginning with the medium condition. The panel on the left of the first row shows the results for true number of features equal to 4. When

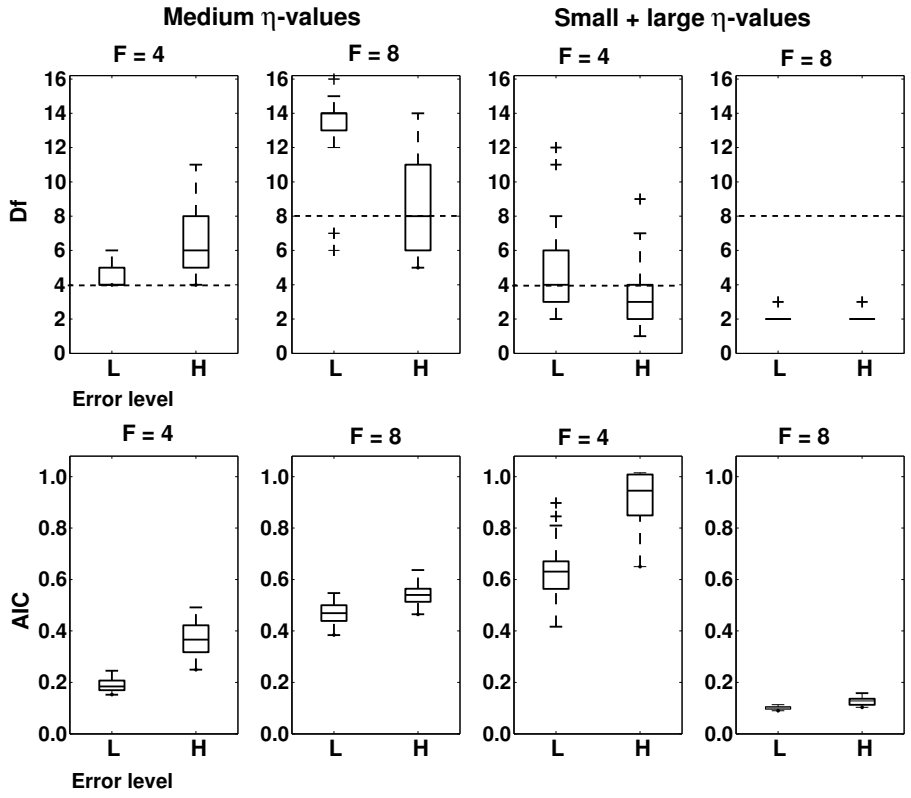

Figure 4.7: Boxplots showing the distributions of 50 simulation samples on 12 objects using the complete set of Gray codes. The experimental conditions are medium (left panels) and small + large (right panels) η values, two error conditions, low (L) and high (H), and two levels of true number of features (4 and 8) corresponding to two levels of n/T ratio equal to 16 and 8. The top panels show the effective number of features selected for each sample (= *Df*) with the true number of features represented as a dashed line. The lower panels show the associated AIC_L values.

error is low, the true number features are well recovered by the Positive Lasso, with a little overfitting for a small proportion of samples. The high error condition clearly shows some overfitting, because most of the models selected by the Positive Lasso contain 6 features. The corresponding AIC_L values show that prediction error is lower when error is low. The next panel shows the results for true number of features equal to 8, meaning that there are more features compared to the number of observations than in the previous condition of 4 features. In the low error condition, there is a considerable amount of overfitting because most of the selected models contain 14 features. The high error condition shows that most of the selected models contain the right number of 8 features. The associated AIC_L values show that the prediction error is higher in the high error condition, and that, despite the overfitting in the low error condition, the prediction error is still very acceptable.

The results for the small + large η values show a different pattern. In the model with 4 features, most of the samples in the low error condition recover models with 4 features, but some overfitting is clearly present. Most of the models selected in the high error condition, have fewer than 4 features. The prediction error is lower in the low error condition than in the high error condition. In the model with 8 features, the selected models all have 2 features, regardless of the error level. This substantial amount of underfitting does not have an effect on the prediction error, which is very low in both conditions. Combining these results with the findings in Table 4.8, we know that, in almost all the samples, the two features with the highest η values are selected.

Summarizing, the true number of features are best recovered in the 4 features model, for both medium and small + large η values. The 8 features model, shows some overfitting for the medium η values, and some underfitting for the small + large η values. In all conditions, the prediction error is satisfactory.

Simulation results on random sample of Gray codes + Filter

The model based on 12 objects allows for comparing the strategy of taking a random sample of features from all possible Gray codes combined with the use of a filter with the strategy of using the whole set of possible distinctive features obtained with all possible Gray codes. To assess the performance of the random sample strategy, the same simulation samples for the 12 objects used with the complete set of distinctive features, were analyzed again using the random sample strategy. The results of the random sample strategy are displayed in Figure 4.8. Since the same samples are used, Figure 4.8 can be compared directly with the results of Figure 4.7 that is based on the complete set of distinctive features.

The results in Figure 4.8 show that in the majority of the experimental conditions, the number of features selected by the Positive Lasso are equal to, or very close to the number of features in the true model. The best results in terms of recovered number of features are obtained when the true feature parameter values are a combination of small and large values. The best prediction error values occur when the true model has smaller number of features (the 4 features model) for both the medium and the small + large η values. When the true number of features is not recovered, there is, in general, a tendency towards overfitting. In particular, the condition with medium

η values and 4 features as the true model, shows a considerable amount of overfitting. However, the associated prediction error values are still the lowest of all the experimental conditions. In conclusion, compared to the method using the complete set of distinctive features, the random sample strategy has higher prediction error, but, in general succeeds in finding subsets of features of (about) the same number as the true models, besides a tendency to overfit in some conditions.

The simulation results for the model based on 24 objects provided additional information on the random sample strategy. Figure 4.9 clearly shows that in this case, the number of features selected by the Positive Lasso exceeded the true number of features considerably, even more when the true number of features is equal to 35,

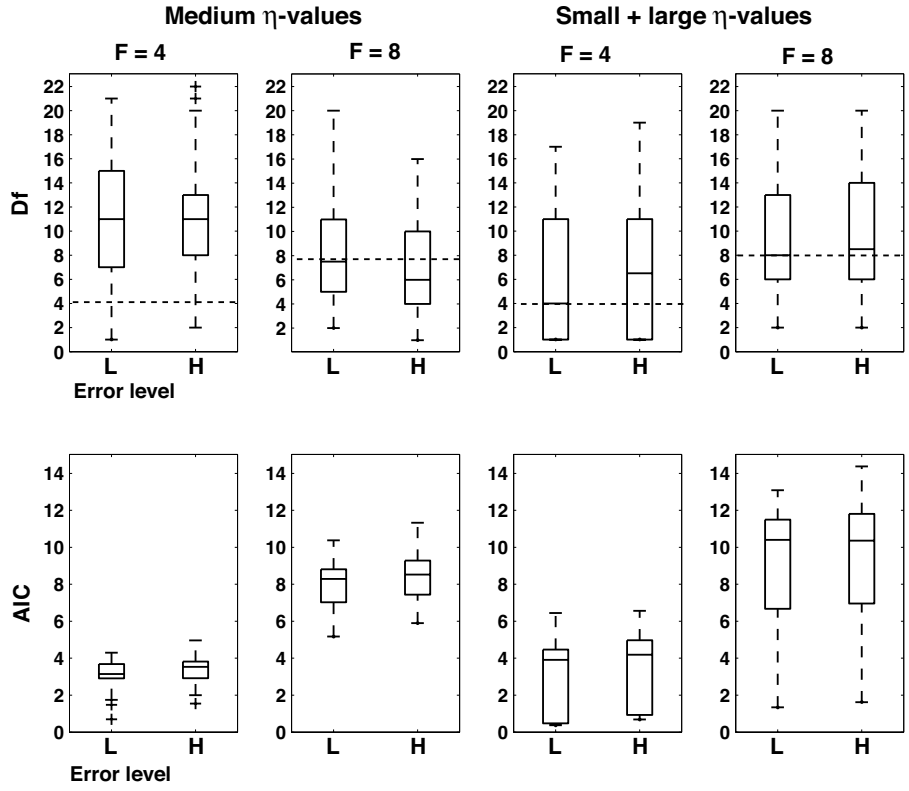

Figure 4.8: Boxplots showing the distributions of 50 simulation samples on 12 objects using a large random sample of the complete set of Gray codes combined with a filter. The experimental conditions are medium (left panels) and small + large (right panels) η values, two error conditions, low (L) and high (H), and two levels of true number of features (4 and 8) corresponding to two levels of n/T ratio equal to 16 and 8. The top panels show the effective number of features selected for each sample (= Df) with the true number of features represented as a dashed line. The lower panels show the associated AIC_{L} values.

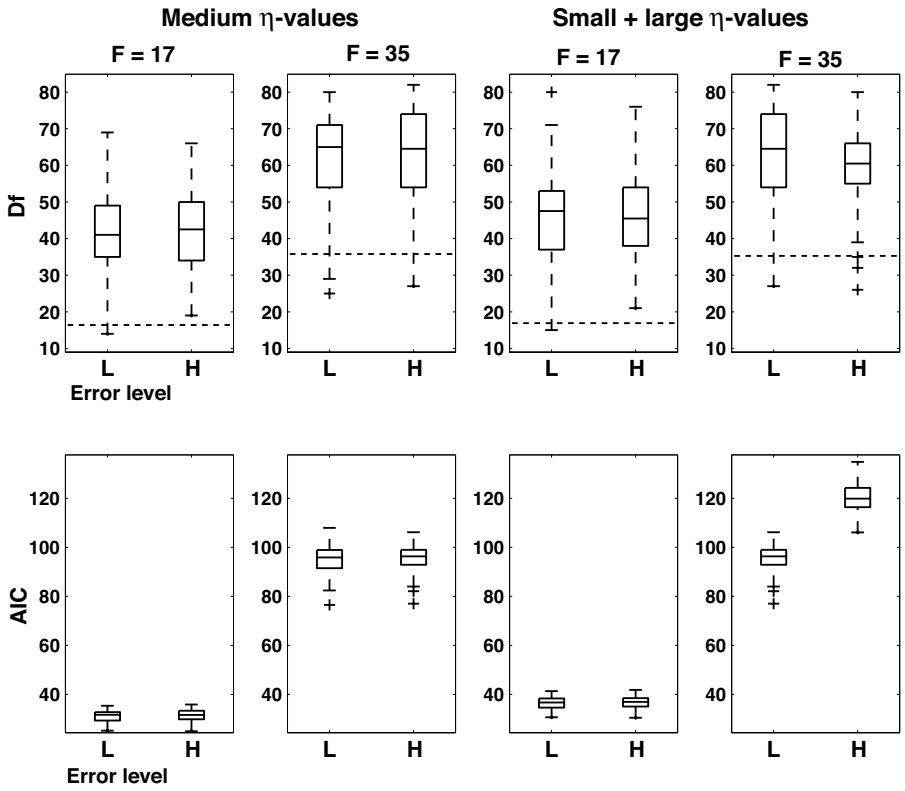

Figure 4.9: Boxplots showing the distributions of 50 simulation samples on 24 objects using a large random sample of the complete set of Gray codes. The experimental conditions are medium (left panels) and small + large (right panels) η values, two error conditions, low (L) and high (H), and two levels of true number of features (17 and 35) corresponding to two levels of n/T ratio equal to 16 and 8. The top panels show the effective number of features selected for each sample (= Df) with the true number of features represented as a dashed line. The lower panels show the associated AIC_L values.

the condition where the number of features is larger compared to the number of observations ($n/T = 8$ condition). The prediction error is much lower for the 17 features model (the condition $n/T = 8$). It is clear that many more features are needed to obtain models with acceptable levels of prediction error.

4.4 Discussion

This paper introduces a method to generate and select a subset of features for the Feature Network Models. It combines feature enumeration using Gray codes with a predictor selection algorithm, the Positive Lasso. The fact that FNM can be considered as a univariate multiple regression problem allows for the use of this type of predictor selection algorithm. The advantages of the multiple regression framework had not been fully explored in earlier feature models related to the FNM. In the following we discuss the two basic elements that constitute our method.

The first element is the enumeration of features with Gray codes. The enumeration of all possible subsets makes use of the adjacency property of the Gray codes, the property that successive numbers differ exactly in one bit. In our study, we did not explicitly use the adjacency property in the feature selection strategy. Instead of exploiting the adjacency property of the Gray codes, we used the codes to efficiently list all possible features where each feature is generated exactly once. The gain of using the Gray code results from its combination with the transformation from the objects $\times$ features matrix $\mathbf{E}$ to the object pairs $\times$ featurewise distances matrix $\mathbf{X}$. This transformation limits the search for predictors to the set of truly distinctive features. To our knowledge, the explicit search in the space of distinctive features has not been used as a predictor generation and selection strategy before. The adjacency property proved to be useful in the simulation study where we needed connected networks to represent the true network configuration. The list of Gray codes for the number of bits equal to the number of features (instead of equal to the number of objects) results in a lattice, or the complete network of all possible feature patterns. Selecting m feature patterns from this matrix of size $[\frac{1}{2}(2^T) - 1] \times T$ ensures a connected network representation for a given number of objects.

In this context, it should be noted that features can be generated using Gray codes in two ways. The first method amounts to generating Gray codes for the number of bits equal to the number of objects. This method is suitable in the situation where there is no a priori knowledge about the possible number of features suitable for the data at hand. Features generated in this way yield a feature matrix of size $m \times [\frac{1}{2}(2^m) - 1]$, which obviously leads to a feature selection problem because there are far more features than observations. The second method is more appropriate for the situation where there is some knowledge available on the number of features that would be reasonable for the data. In that case, features could be generated for the number of bits equal to the number of features instead of equal to the number of objects, resulting in a matrix of size $[\frac{1}{2}(2^T) - 1] \times T$. Again, the symmetric property of the Gray code allows for discarding the second half of the code. It is no longer a feature selection problem because the number of features is known in advance, but rather a problem of finding the right feature pattern for each object, a problem related to the travelling salesman problem. The set of all possible feature patterns given a fixed number of features can be viewed as a lattice, or complete network of all possible feature patterns. The best selection of feature patterns to describe a given number of objects should correspond to one of the possible shortest path routes on this lattice.

For the situation where one searches for sets of fixed numbers, Gray codes have been frequently used to list all p-element subsets of a q-element set in such a way that consecutive sets differ by exactly only one element (*cf.* Nijenhuis & Wilf, 1978). Applying this particular use of Gray codes to the FNM would lead to the complete enumeration of all possible subsets of features. It is well known that the number of subsets grows exponentially with the number of objects m and the number of features, limiting the strategy to a very small number of objects and features (a small explorative study showed that in the context of FNM complete enumeration of all subsets is limited to number of objects $\leqslant 5$ and number of features $\leqslant 9$). Attacking the problem of selecting a subset of features in this way would be an NP-hard problem (NP = nondeterministic polynomial problem, a category of problems that cannot be solved exactly in polynomial time, but for which the verification of the solution can be accomplished in polynomial time). Even if optimal solutions could be obtained, they would not necessarily yield a good compromise between model complexity and prediction accuracy. Since our method attempts to achieve this compromise it is difficult to compare it to conceptually different techniques like the aforementioned and the cluster differences scaling algorithm that has been used earlier in FNM.

Using our method, the generation of all possible distinctive features with Gray codes is feasible for $m \leqslant 22$. Simple binary codes could have been used instead, although, the generation of successive objects that differ in only one bit, might be faster (*cf.* Savage, 1997). If the number of objects exceeds 22, features are generated by taking a very large sample from the whole set of Gray codes followed by a filter technique that selects the features with the highest separate discriminability parameters. The results of the simulation study show that in this case, the number of features selected by our method exceeds the true number of features in some conditions. Therefore, feature generation for $m > 22$ must be further improved.

Given a very large set of features, obtained with the complete list of Gray codes or a large sample of this list, the second element of our method consists of selecting the best subset of features using the Positive Lasso, an adaptation of the Lasso algorithm to meet the positivity constraints of FNM. The results obtained with the Positive Lasso are in accordance with results obtained with the Lasso: the Lasso tends to perform best with a combination of small and large parameter values (Friedman & Popescu, 2006; Tibshirani, 1996). The results of our simulation study shows that the best results are obtained in the condition formed by the combination of small and large feature discriminability parameters. The Positive Lasso is also useful in the situation of features given by theory or provided by the experimental conditions, and helps to select the relevant features. We used the Positive Lasso as a tool, but it certainly merits to be studied in its own right since all its properties are not exactly known yet.

To select the amount of shrinkage, we used an AIC-like criterion adapted for the Lasso context. It is well known that AIC, in contrast to BIC, has a tendency towards overfitting (*cf.* Zou et al., 2006) and this property probably explains part of the overfitting observed in our simulation study. However, it is also known that AIC and BIC possess different asymptotic optimality (*cf.* Zou et al., 2006): if the true regression function is not in the candidate models, the model selected by AIC

asymptotically achieves the smallest average squared error among the candidates. BIC on the other hand is known for its consistency in selecting the true model: if the true model is among the candidate models, the probability of selecting the true model with BIC approaches 1 as the sample size approaches infinity. Given our setting, when the number of objects exceeds 22 and as a consequence we have to take a large random sample from the total set of features, we know in advance that the true model might not be present among the candidate models, which motivates the choice for the AIC criterion. Yang (2005) recently explored the possibilities of combining both strengths of BIC and AIC in the context of regression estimation and concluded that there are some theoretical and empirical results in support of adaptive model selection, but that it is still not clear whether it can really combine the strength of AIC (= prediction optimality or minimax-rate optimality) with the strength of BIC (consistency).

The combination of the two elements (the enumeration of features with Gray codes and the selection of the best subset of features) leads to a method that aims at selecting a subset of features that constitutes a good compromise between model fit and model complexity. The Gray codes allow for defining a finite solution space, which can be further reduced by restricting the search to the distinctive features only. In fact, features are generated from a model instead of being the result of collecting empirical data. Next, the Positive Lasso algorithm selects the best subset regardless of the number of features in the set. Instead of searching for the optimal solution in the distinctive features space, we prefer a suboptimal solution, selected with the Positive Lasso, that has better generalizability properties.

The method described in this paper can be applied directly to the common features model, with one restriction. Given that the total set of common features is larger than the total set of distinctive features, the limits of complete enumeration of the set of common features will be reached earlier than with 22 objects, the limit for the distinctive features model.

Chapter 5

Network Representations of City-Block Models [1]

Abstract

City-block models for similarity always allow network representations that reproduce the same distances as the unique coordinate representation. A rule to construct such networks is given, based on additivity of city-block distances across sequences of intermediate points along monotonic trajectories in space. The paper also defines the concept of internal node, which helps in reducing the complexity of networks and in making them better interpretable. The general graph construction rule and definition of internal nodes also apply to the distinctive features model, the common features model (additive clustering), as well as to hierarchical trees, additive trees, and extended trees. Additivity is the key property that makes the city-block metric so versatile and causes a basic unity of dimensional, hierarchical and featural representations of similarity.

5.1 Network representations of city-block models

The city-block distance rule has been under consideration in psychology as a plausible model for similarity and difference for a long time (Arabie, 1991; Attneave, 1950; MacKay, 2001; Micko & Fischer, 1970; Nosofsky, 1984; Shepard, 1964). It has been used not only for human perception (Borg & Leutner, 1983; Garner, 1974; Shepard, 1987), but also for category learning (Kruschke, 1992; Zaki, Nosofsky, Stanton, & Cohen, 2003), color vision and pattern recognition in honeybees (Backhaus, Menzel, & Kreißl, 1987; Ronacher, 1992), as well as for perception of electric properties of objects by weakly electric fish (Emde & Ronacher, 1994). The model has caused a flux of technical papers concerned with the computational complications that arise when trying to fit city-block distances to error-contaminated (dis)similarity data (Brusco, 2001, 2002; Eisler, 1973; Eisler & Roskam, 1977; Groenen & Heiser, 1996; Groenen,

[1] This chapter has been submitted for publication as: Heiser, W. J. & Frank, L. E. (2005). Network representations of city-block models. *Submitted manuscript.*

Heiser, & Meulman, 1998, 1999; Heiser, 1989, 1991; Hubert & Arabie, 1988; Hubert, Arabie, & Hesson-Mcinnis, 1992; Okada & Imaizumi, 1980). There are other unsolved technical problems; for example, degeneracies in nonmetric multidimensional scaling with all distances tied into only two values are more prevalent in the city-block metric (and in the dominance metric) than in other Minkowski metrics (Shepard, 1974). In this paper, we leave these technical issues aside, and focus primarily on some theoretical properties that lead to equivalent representations of the city-block model.

Substantively, the city-block model has played a major role in the classic distinction between integral and separable stimulus dimensions, which is an essential consideration in most current experimental and theoretical analyses of category learning (Ashby & Maddox, 1990; Goldstone, 1994; Kruschke, 1992; Melara, Marks, & Lesko, 1992; Nosofsky, 1992). Shepard (1964) reported two experiments specifically designed to test if the metric of psychological space depends on the perceptual analyzability of the stimuli, and found that for objects differing in size and angle of orientation the city-block distance gave a better account of subjective judgments of similarity and objective measures of generalization than the Euclidean distance. Together with results on category learning (Shepard & Chang, 1963; Shepard, Hovland, & Jenkins, 1961), these findings also demonstrated a fundamental role of selective attention for analyzable stimuli (Shepard, 1991). This line of research culminated in the generalized context model for category learning and attention allocation (Nosofsky, 1984, 1986, 1987, 1992; Nosofsky & Zaki, 2002; Zaki et al., 2003).

Closely connected to the integrality-separability distinction is the uniqueness of the coordinate system. In the words of Attneave,

> "One possible hypothesis would be that the psychological dimensions are related like physical dimensions in Euclidean space. Another would be that differences along different dimensions combine additively, in which case composite judgments would be predicted by a multiple linear regression equation. Perhaps the most significant psychological difference between these two hypotheses is that the former assumes one frame of reference to be as good as any other, whereas the latter implies a unique set of psychological axes." (Attneave, 1950, p. 555).

One way to distinguish between integral and separable dimensions is to establish whether a stimulus is more readily associated with another stimulus that is close to it in the Euclidean metric or with one that may be farther away but matches it on some pre-determined dimension. Various other converging operations have been used to distinguish between these two types of dimensions (Garner, 1974). The unique coordinate system of the city-block metric has also motivated other utilizations. Buja and Swayne (2002) used dimensional uniqueness to identify an orientation of Euclidean solutions, which are rotationally invariant. Heiser (1989) used dimensional uniqueness as an argument to develop an individual differences city-block model with dimension weighting.

Nevertheless, uniqueness and additivity of city-block dimensions do not tell us what structural relations are valid in the whole space. For example, uniqueness and additivity do not tell us if the stimuli are clustered or not, whether two stimuli are

close neighbors or not, and whether three stimuli have the same order on all dimensions or not. It is remarkable that in applications of the city-block model, there has not been much attention for actual representations. Some authors do not even show or list the coordinates; they only report tests of inter-dimensional additivity, or the relative goodness-of-fit (Melara et al., 1992; Ronacher, 1992; Emde & Ronacher, 1994). One reason for this lack of attention for coordinates might be that the psychological dimensions are supposed to be monotonic with physical dimensions present in the stimuli, so that the order of the coordinates is known by design. However, inter-dimensional additivity does not preclude the possibility that stimulus differences along one dimension change for different levels of the other dimension, i.e., that they show non-linear structural relations (as will become clear in an example of similarity between rectangles that is discussed in the following).

The present study was triggered by the notion that a simple and direct way to describe structural relations between objects is to draw a network, with nodes (or vertices) for the stimuli and with lines (or edges) indicating local connections between neighbors. Distance in a network is the length of the (shortest) path traveled. If the stimuli are clustered, we expect to find fully connected subsets, or cliques. If three stimuli have the same order on all dimensions, we expect to find segmented pathways without sharp turns. If there is interaction between dimensions, we expect a nonlinearly distorted grid, and so on. Would it be possible to use the rectangular grid that is so characteristic for the city-block metric and just connect all pairs of points on the grid whenever there is no other point lying between them, and finish by dropping the rest of the grid? Would it still be possible to reconstruct the distance correctly if we replaced all city-block corners by direct straight lines?

It turns out that it is indeed possible to develop a universal network representation of city-block models that applies regardless of the dimensionality of the coordinate space. This paper first describes the key elements of the network construction method, which are the concepts of *betweenness*, *metric segment*, and *metric-segmental additivity*. Since a network is just a collection of nodes and lines, one needs some embedding to be able to draw it, but the details of this embedding are of secondary importance. While the network is the model, an embedding is one of several possible maps of it. The paper also introduces the possibility of including an additional set of points corresponding to hypothetical stimulus objects, called *internal nodes*. An example of the perception of rectangles will demonstrate their use. It is shown that networks throw new light on a puzzling characteristic of the city-block model, the occurrence of partial isometries. Next, the same theory is applied to the Goodman-Restle symmetric set difference, a special case of the city-block metric, with binary dimensions called distinctive features. This framework contains a rather large class of discrete models for similarity data, including additive similarity trees (Buneman, 1971; Sattath & Tversky, 1977), extended similarity trees (Corter & Tversky, 1986), the additive clustering or common features model (Carroll & Arabie, 1983; Shepard & Arabie, 1979), and a new set-partitioning model with unicities called the *double star tree*. It is shown that the same network construction rule recovers the familiar additive tree graph, and yields new graphical representations for the other models. These are illustrated with several examples of similarity data known from the literature.

5.2 General theory

The discussion starts with the fundamental notion of betweenness in continuous spatial models, and demonstrates how it leads to additivity of distance. Betweenness in more dimensions requires the concept of a metric segment, which is the area between a pair of points that contains all intermediate points for which distance is additive. Then a network representation is formed from a complete network by elimination of lines when additivity applies. Some properties of this representation are discussed with an example of similarity between rectangles. Next, the concept of an internal node is introduced as a supplementary point located in the metric segment of any two objects, or in the intersection of several metric segments. This section concludes with a general characterization of partial isometry, a problematic phenomenon that is specific for the continuous city-block model.

Betweenness of points and additivity of distances

Geometric models like the city-block model consist of points arranged in some continuous space, among which we define distances according to a certain rule (or metric) to account for empirical relations between the experimental objects. An elementary structural property of spatial arrangements is the *betweenness relation*. In some situations, betweenness implies additivity of distance. Taking the simplest case, when we have three ordered points A, B, and C in one dimension, where B is between A and C, the distance between the outer points A and C is the sum of the distances from A to B and from B to C. In other words, when B is between A and C on a line, a condition called *intra-dimensional betweenness*, we have *intra-dimensional additivity*. It is easy to see that more generally, the distance between any two points on a line is equal to the sum of the lengths of the segments that one crosses when going from one to the other through a series of intermediate points.

In more than one dimension, the situation changes because it is a common characteristic of all metrics that distances satisfy the triangle inequality. Denoting the distance between two points A and B by $d(A, B)$, the triangle inequality states that $d(A, C) \leqslant d(A, B) + d(B, C)$. Therefore, going through a third point can only add to the distance. Even if an intermediate point B is between two others on all dimensions, in going from A to C the direct route is generally shorter than going via B. In Euclidean space, the only exception is when three points are located exactly in a one-dimensional subspace, in which special case the triangle inequality reduces to an equality (Torgerson, 1952). By contrast, in city-block space, the triangle equality is much more common, because betweenness in all city-block dimensions (a condition that we will call *metric-segmental betweenness*) always leads to additivity of distance.

We now demonstrate the particular result that under the city-block metric the triangle inequality reduces to an equality for any three points A, B, and C whenever B is between A and C on all dimensions (Busemann, 1955, p. 28). Let A have coordinate values z_{At}, for $t = 1, \cdots, T$ where T denotes the number of dimensions. The city-block distance between A and B is defined as the function

$$d(A, B) \quad = \quad \sum_t |z_{At} - z_{Bt}|. \tag{5.1}$$

The fact that $d(A, B)$ is built up as a sum of dimension-wise differences is called *inter-dimensional additivity* (Suppes, Krantz, Luce, & Tversky, 1989, section 14.4.3). For metric-segmental additivity to hold, the coordinates have to satisfy, for each dimension t, either $z_{At} \leqslant z_{Bt} \leqslant z_{Ct}$ or $z_{At} \geqslant z_{Bt} \geqslant z_{Ct}$ (*monotonicity*: all choices of z_{Bt} within the constrained area lead to monotonically increasing or monotonically decreasing sets of coordinate values). Under monotonicity we must have, for any t,

$$(z_{Ct} - z_{At}) = (z_{Ct} - z_{Bt}) + (z_{Bt} - z_{At}), \tag{5.2}$$

$$|z_{At} - z_{Ct}| = |z_{At} - z_{Bt}| + |z_{Bt} - z_{Ct}|, \tag{5.3}$$

where the three terms in Equation 5.2 are either all positive or all negative, so that we can take absolute values and freely reverse the order of the arguments in Equation 5.3, which expresses intra-dimensional additivity for any dimension. Summing Equation 5.3 over t and using Equation 5.1 we obtain

$$\begin{aligned} \sum_t |z_{At} - z_{Ct}| &= \sum_t |z_{At} - z_{Bt}| + \sum_t |z_{Bt} - z_{Ct}|, \\ d(A, C) &= d(A, B) + d(B, C), \end{aligned} \tag{5.4}$$

that is, *metric-segmental additivity* of distance when we go from A to C via B. This result forms the basis of the network representations that we develop in this paper. Joly and Le Calvé (1994) have defined the general concept of a *metric segment* as the set of points $[AB]_{met} = \{M\colon d(A, B) = d(A, M) + d(B, M)\}$. The metric segment is a generalization of the line segment to multidimensional spaces. In Euclidean space, metric segments are still segments of lines, but in two-dimensional city-block space, they are rectangles with sides parallel to the axes. In three-dimensional city-block space, metric segments are *cuboids* (parallelepipeds with rectangular faces), and in more than three dimensions, *hypercuboids*. When dimension-wise differences are all equal, these structures reduce to squares, cubes and hypercubes.

The prevalence of metric-segmental additivity in city-block space simply expresses the fact that in this type of space, there is a multitude of paths through intermediate points covering exactly the same distance. As every passenger knows, there is a unique shortest route by air from city to city, but within any city where buildings are arranged in rectangular blocks one can reach distant destinations along several different routes that are equally long.

Network representation of city-block configurations

The surprising consequence of metric-segmental additivity is that it allows us to construct a model representation of city-block configurations that does not involve coordinate values. This coordinate-free representation consists of a set of *nodes* or *vertices* $\mathcal{V} = \{\nu_1, \cdots, \nu_i, \cdots, \nu_m\}$, representing the objects, and a set of line segments or *edges* $\mathcal{T} = \{\tau_1, \cdots, \tau_l, \cdots, \tau_L\}$, where $L \leqslant \frac{1}{2}m(m-1)$, connecting pairs of nodes. Each edge τ_l has a length q_l, collected in the set $\mathcal{Q} = \{q_1, \cdots, q_l, \cdots, q_L\}$, which indicates the distance between the corresponding pair of nodes. Thus, the triad $\mathcal{N} = \{\mathcal{V}, \mathcal{T}, \mathcal{Q}\}$ forms a valued graph or *network*. In a full, or *complete* network, we have $L = \frac{1}{2}m(m-1)$, that is, al $n = \frac{1}{2}m(m-1)$ pairs of nodes are connected

by an edge. Of course, in applications, we would like simplicity to prevail by aiming at an *incomplete network* with L as small as possible (Klauer & Carroll, 1989). If $L < (m-1)T$, the network is a more parsimonious parametrization than the coordinate space. Moreover, as we shall see shortly, there are other considerations that also can make a graphical representation attractive.

The construction of the network goes as follows. Given a city-block configuration $\mathcal{M} = \{\mathbf{Z}, \mathbf{D}\}$, where $\mathbf{Z}$ is a $m \times T$ matrix of coordinates z_{it}, with T the number of dimensions, and $\mathbf{D}$ a matrix of distances $d(Z_i, Z_j)$ between pairs of points Z_i and Z_j, the set $\mathcal{V}$ is formed by just allocating a separate node ν_i to each distinct Z_i. The set $\mathcal{T}$ is then formed by elimination. We start with a full list of edges, with the distances listed in some fixed order in $\mathcal{Q}$. For all triads of points Z_i, Z_j, and Z_k, we determine the quantity $W_{ik}^j = d(Z_i, Z_j) + d(Z_j, Z_k) - d(Z_i, Z_k)$. Then Z_j belongs to the metric segment $[X_i X_k]_{met}$ if $W_{ik}^j = 0$. When there is at least one Z_j for which $W_{ik}^j = 0$, we can drop the direct edge from ν_i to ν_k from the list $\mathcal{T}$, while keeping the direct edges from ν_i to ν_j and from ν_j to ν_k. In that case, we also omit the corresponding distance from the list $\mathcal{Q}$. Dropping the direct edge is possible since metric-segmental additivity transfers to additivity along the shortest path in the graph. Thus, we are able to use an interesting parallel between the coordinate space and the graphical space. While the city-block distance between a pair of points is equal to a sum of distances through a series of intermediate points in their metric segment, the graphical distance is equal to the sum of edge lengths in a shortest path that connects two nodes. When $W_{ik}^j \neq 0$ for all i, j, k, no edges are dropped and the complete network $\mathcal{N}$ trivially represents $\mathcal{M}$.

There is a caveat for the particular case in which two distinct objects, i and j, have the same location, $Z_i = Z_j$. Then $d(Z_i, Z_j) = 0$ and $d(Z_j, Z_k) = d(Z_i, Z_k)$, from which it follows that $W_{ik}^j = 0$, so that the direct edge between ν_i and ν_k is dropped. The same equalities also give $W_{jk}^i = 0$, with the effect that the direct edge between ν_j and ν_k is dropped. Consequently, two objects with the same location would become two nodes that are disconnected from all other nodes in the graph (isolates). By merging such objects into one node, which has the same distances to the other points, and again determining the relevant metric segments, the graph will generally become connected.

The graph $\mathcal{N}$ by itself is the desired network representation. However, to visualize or interpret $\mathcal{N}$, we must embed it again in some coordinate space. Note that we now have more freedom in choice of embedding, since the primary elements of interpretation are the connectivity and structural order relations between the nodes, while the exact length of the edges is secondary. The embedding may be in the original city-block space if it is two-dimensional, or in some other space with two dimensions. We could use a Euclidean embedding of the graph, obtained, for instance, by a nonmetric MDS method (*cf.* (Buja & Swayne, 2002)), or by a metric MDS with weights to down-weight the large graphical distances (Kamada & Kawai, 1989). Of course, we can reconstruct the original city-block distances $\mathbf{D}$ only if the edges included in the plot of the embedding are precisely those from the list $\mathcal{T}$, labeled with the edge lengths $\mathcal{Q}$. If we would use any other common procedure to draw lines

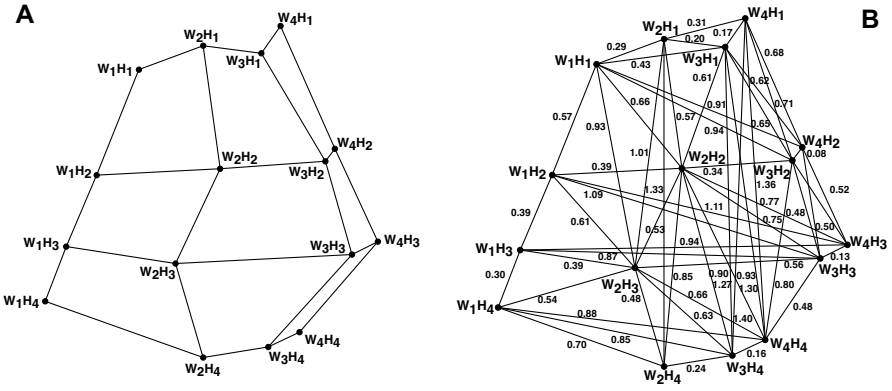

Figure 5.1: City-block solution in two dimensions for the *rectangle* data. The labels $W_1 - W_4$ indicate the width levels, and $H_1 - H_4$ the height levels of the stimulus rectangles.

between pairs of objects in an embedding, for example, by determining a threshold graph or a K-nearest neighbor graph (*cf.* Jain & Dubes, 1988, p. 60), reconstruction of the original distances by their counterparts in the graph is generally inaccurate.

To illustrate the graphical representation of a city-block configuration, we look at some data collected and analyzed by Borg and Leutner (1983). The stimuli used were 16 rectangles of varying width and height, where the two variables each had four levels increasing in equal steps. All 120 possible pairs of stimuli were presented twice, in random order, to 21 subjects, who had to rate each pair on a 10-point scale of dissimilarity. Reliability, calculated per subject as the product-moment correlation over the ratings of stimulus pairs in the two different orders, was 0.75 on average. The data, averaged over all subjects and replications, were analyzed in two dimensions[2] with the smoothing method for city-block multidimensional scaling described in Groenen et al. (1998). This method was specifically designed to avoid being trapped in local minima of the least squares MDS criterion. Figure 5.1 gives two versions of the two-dimensional solution. In both versions, the points are labeled with their width level and their height level. Thus, W_1H_1 (top-left) is the smallest rectangle, and W_4H_4 (bottom-right) the largest. In Figure 5.1A, we have connected the points with their direct neighbors by design; that is, lines connect rectangles differing one level on only one variable (as in Borg & Leutner, 1983, their Figure 3). It shows that the horizontal dimension roughly corresponds to width, the vertical dimension to height, and that the intervals tend to become smaller as the size of the rectangles increases, in both dimensions. Borg and Leutner predicted this nonlinear effect on psychophysical grounds; it was also present in their solution. However, contrary to their solution, the current solution also exhibits interaction: successive width intervals tend to become larger as height levels increase, although not uni-

[2]The fitting criterion used was least squares and metric, since Borg and Leutner (1983) reported that non-metric fitting showed a linear relationship between dissimilarity and distance.

formly.

Figure 5.1B gives the network representation, using the same coordinates. Here, two points are connected if there are no other points between them, that is, if their metric segment is empty. Recall that in a two-dimensional city-block solution, a metric segment has a rectangular shape, with orientation parallel to the horizontal and vertical axes. The two points spanning the metric segment are on a main diagonal of that rectangular area. This main diagonal is shown as a line in Figure 5.1B if the metric segment contains none of the other points. For example, a line connects W_1H_4 and W_2H_3, since they span an empty metric segment. One could also say that the two candidates in the design for being inside their metric segment, W_1H_3 and W_2H_4, are actually located outside of it. Thus, while Figure 5.1A emphasizes conformities of the solution with the design, Figure 5.1B highlights violations as well. Also, note that even though a path like $W_1H_1 - W_1H_2 - W_1H_3 - W_1H_4$ is in conformity with the design, the tilting to the right contradicts that these stimuli are of equal width. However, this contradiction shows up as a property of the spatial city-block solution with the horizontal dimension identified as width, not as a property of the network, which could have been plotted differently. Furthermore, a path like $W_1H_4 - W_2H_4 - W_3H_4 - W_4H_4$ is correctly monotonic in the horizontal direction; yet it has two subadditivities giving direct lines from W_1H_4 to W_3H_4 and W_4H_4. The total number of lines in Figure 5.1B is 56, while the spatial solution has 30 independent coordinates. Therefore, the network representation is not parsimonious, but it does enable a detailed analysis of structural relations in the data.

It might appear that the network representation is unduly complex, compared to the simplicity of the spatial representation, in which we just plot the coordinates. More specifically, the network seems to have the following unfavorable properties:

1. Some relatively long lines appear in Figure 5.1B, e.g. between W_2H_1 and W_2H_4 or between W_1H_3 and W_4H_3. By contrast, the attraction of other network models often is that they have global properties resulting from the action of local connections (short lines). Here, the long lines simply reflect that objects can be opposites on one dimension and direct neighbors on the other dimension.

2. Many nodes have high degree (number of lines that are incident with it, or number of nodes adjacent to it); for example, 11 lines emanate in Figure 5.1B from node W_2H_2, and 10 lines from W_2H_3, while the lowest degree still is 5 (for W_1H_3, W_1H_4, and W_2H_4). As can be seen from Figure 5.1A, the current design predicts nodes with degree 2, 3, or 4.

3. Many crossings of lines occur at locations where there is no intermediate node. For example, the line between W_2H_1 and W_2H_4 in Figure 5.1B crosses 15 other lines without meeting any other node. In the design of Figure 5.1A, these rectangles are connected via the much simpler three-segment path $W_2H_1 - W_2H_2 - W_2H_3 - W_2H_4$.

4. The total number of lines is large, 56. If we would consider each line length as a separate parameter, the network model absorbs many parameters, compared to the number of independent data values (120). However, it should be noted

that we actually fitted only $2 \times (m-1) = 30$ coordinate values, so the line lengths cannot be considered as independent quantities. In the design, we have only 24 lines, and under a model of no interaction, these line lengths would be further constrained to six independent parameters (3 width intervals and 3 height intervals).

Since these properties are also prevalent in other examples that we have analyzed but do not report in detail here, they seem to be a recurring and genuine characteristic of the present network representation. However, as shall become clear in the next section, there is a way to alleviate points 1-3, and to some extent point 4 as well, by introducing an additional set of points, called *internal nodes*, which also play an important role in special cases of the model.

It is of special interest to look at the one-dimensional case. If $\mathbf{D}_1$ is a distance matrix of points along a line, then any point lies between two others except for the endpoints, from which it follows immediately that the graph $\mathcal{N}$ has exactly $m-1$ edges. Assuming the rows and columns of $\mathbf{D}_1$ are ordered in the same way as the order of the points along the line, the edges of the graph correspond to the elements on the subdiagonal of $\mathbf{D}_1$. We find segmental additivity for all pairs of points (i, j) that are not consecutive. Specifically, any other element in the upper-right triangle of the distance matrix $\mathbf{D}_1$ is the sum of consecutive elements in the subdiagonal, starting with the subdiagonal element in the same row, and ending with the subdiagonal element in the same column. Hence, in this case the graph is a chain, which has graphical distances with exactly the same additivity structure as a set of points on a line. The chain can be displayed in many ways (for instance, as a curved, connected sequence of nodes in the plane), all of which give an equivalent reconstruction of the one-dimensional distances, as long as their edge lengths are equal to the subdiagonal elements of $\mathbf{D}_1$.

Internal nodes

It can be useful to add nodes to the network that do not correspond to the original set of points in the city-block configuration $\mathcal{M}$. These additional nodes are called *internal nodes*, and can be chosen in a number of ways. In general, adding one point to a network of m nodes leads to m additional edges in the network. Therefore, the introduction of the internal node should entail the possibility of dropping a number of edges, too. By placing the new point in a metric segment of a pair of existing points, the total number of edges reduces by one. Thus, the internal point could be chosen so that it is in the intersection of as many of the n metric segments as possible.

The case for which the greatest simplification occurs is an equal-distance configuration $\mathbf{Z}_0$, for four points defined as:

$$\mathbf{Z}_0 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{5.5}$$

It is not hard to verify that the city-block distances between all six pairs of points in $\mathbf{Z}_0$ are equal to 2, and that no point is in the metric segment of any other two

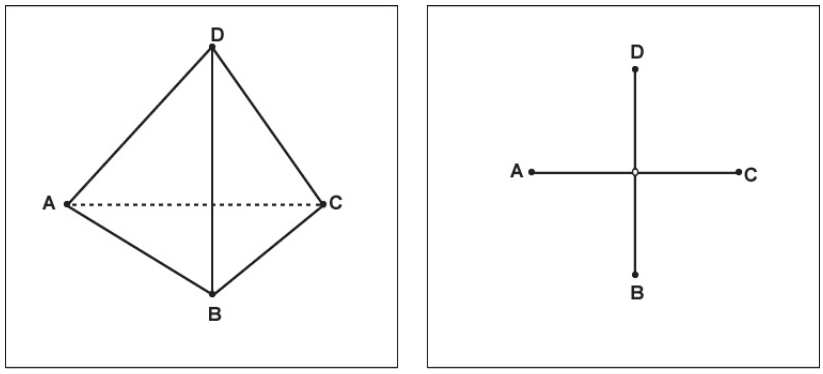

Figure 5.2: Equal city-block distances among four points. Tetrahedron with equal edge lengths (*left panel*) and star graph with equal spokes, which generates the same distances (*right panel*).

points[3]. Hence, the network for $\mathbf{Z}_0$ is a complete graph with equal edge lengths, a structure called a *simplex*. Now, note that the intersection of all metric segments in $\mathbf{Z}_0$ contains exactly one point, the origin $[0\ 0]'$. Introducing the origin as an internal node, four edges are added to the network, all with edge length 1, which brings the total number of edges up to 10. However, since the origin is in all of the six metric segments, the six original edges can all be dropped, bringing the total number of edges down to four. Figure 5.2 shows the simplex and the reduced network. In general, in the equal-distance case the number of edges can always be reduced from $\frac{1}{2}m(m-1)$ to m by the introduction of one internal node. The resulting graph is a special case of a *star graph* (Carroll, 1976) and the resulting metric is called a "center distance" (Le Calvé, 1985).

Let us describe the star graph and the center distance in general terms, as they are a special city-block structure of independent interest; we will encounter this case again later. First, a special four-dimensional configuration yields the same city-block distances as $\mathbf{Z}_0$ in Equation 5.5. It is just the uniform diagonal matrix $\mathbf{Y}_0$ defined as

$$\mathbf{Y_0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.6}$$

The uniform diagonal configuration $\mathbf{Y}_0$ in Equation 5.6 can be generalized to arbitrary m and unequal distances, whereas $\mathbf{Z}_0$ in Equation 5.5 cannot. In particular, collecting a set of object-specific, non-negative weights $\mathcal{A} = \{\alpha_1, \cdots, \alpha_m\}$ as diagonal entries in the $m \times m$ diagonal matrix $\mathbf{Y}$, we calculate the city-block distance

[3]Note that a diagonal matrix with all diagonal elements equal to 2 generates the same set of distances.

between any two rows of $\mathbf{Y}$, denoted by A_i and A_j, as

$$d(A_i, A_j) = \sum_t |y_{it} - y_{jt}| = |y_{ii} - 0| + |0 - y_{jj}| = \alpha_i + \alpha_j, \tag{5.7}$$

where the simplification follows from the fact that $y_{ij} = 0$ if $i \neq j$. Thus, diagonality of a city-block configuration leads to an additively decomposable metric. Although the distance function has additive form, note that for $i = j$ we have $d(A_i, A_i) = 0$, and *not* $2\alpha_i$. Therefore, the distance matrix $\mathbf{D}$ is not additive. We can choose between two geometrical representations of Equation 5.7: either as a polytope with m vertices in $m - 1$ dimensions, which follows from the geometry of the rows of $\mathbf{Y}$, or as a star graph with m external nodes or *leaves*, one internal node or *hub* (corresponding to the m-vector of zeros), and m edges or *spokes*. The spokes have the special property that they all coincide in the hub, and are of length α_i. The regular simplex shown in the left panel of Figure 5.2 is a four-point polytope with edges of equal length, while the reduced network in the right panel of Figure 5.2 is a star graph with four leaves, one hub, and four spokes of equal length.

We now return to the general case to demonstrate the use of internal nodes. The two-dimensional city-block solution for the rectangle data of Borg and Leut-

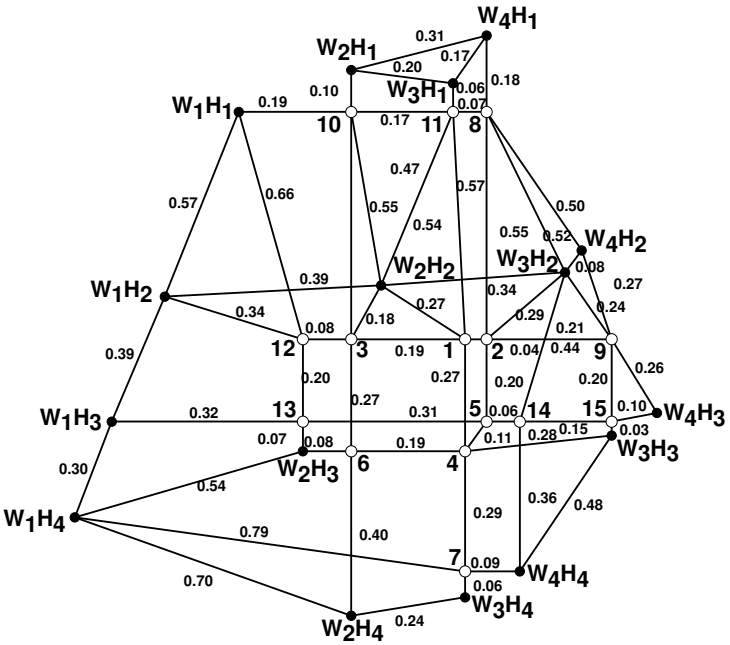

Figure 5.3: Network representation of the two-dimensional city-block solution for the *rectangle* data, including fifteen internal nodes. The labels $W_1 - W_4$ indicate the width levels, and $H_1 - H_4$ the height levels of the stimulus rectangles.

ner (1983), discussed earlier in connection with Figure 5.1, is plotted as a network with internal nodes in Figure 5.3. The internal nodes are indicated with open dots, and labeled according to the order in which they were created, while the external nodes (or leaves) are indicated with solid dots and labeled with their width and height level. The introduction of internal nodes 1 and 2 eliminated all long lines in Figure 5.1B between W_3H_1 and W_4H_1 on the one hand, and W_3H_4 and W_4H_4 on the other hand. Similarly, the introduction of internal node 3 eliminated the long lines between W_2H_1, W_2H_2, W_2H_3 and W_2H_4, all rectangles of width 2. This strategy was continued for all rectangles of height 3 and other subsets until, starting with internal node 7, new nodes were introduced with the additional objective of reducing the degree of the external nodes and the number of crossings. The result in Figure 5.3 more clearly shows the city-block character of the solution than Figure 5.1B, while still accounting for the same distances. It may be verified that if the shortest path between two points includes one or more internal nodes, their direct distance in Figure 5.1B equals the sum of the path lengths in Figure 5.3 (up to rounding error).

After adding 15 new nodes, the total number of lines has a small increase from 56 to 61, of which 20 are among internal nodes only, 28 are between internal and external nodes, and 13 are among external nodes only. The longest lines have been eliminated, and all nodes have lower degree. For example, node W_2H_2 now has degree 6, while it had degree 11 before, and node W_2H_3 now has degree 3, while it had degree 10 before. In addition, the number of crossings at locations without intermediate node has decreased considerably. Thus, internal nodes can indeed simplify several aspects of the network representation, and can make it readily interpretable.

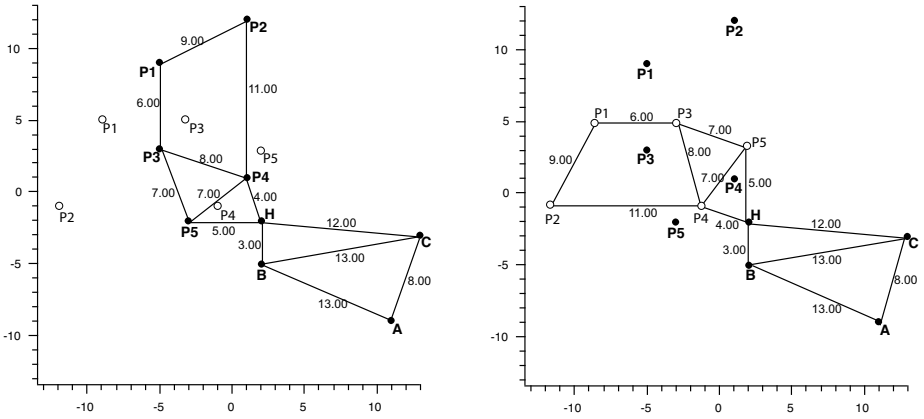

Figure 5.4: Partial isometry: two different configurations with the same city-block distances. *Left panel*: Network representation of A, B, C and the points P1−P5. *Right panel*: Network representation of A, B, C and the points *P1−P5*. The two networks share the internal point H, the hub.

Partial isometries

The network representation helps to explain a puzzling phenomenon that can occur in city-block models. Bortz (1974) has noted that under certain conditions the model coordinates are not unique over and above the usual indeterminacies in distance models, such as invariance of distances under translation (or choice of origin) and reflection of dimensions. Figure 5.4 shows an example (adopted from Bortz, his figure 3), in which two city-block solutions $\mathcal{M}$ and $\mathcal{M}'$ are superimposed: they have the points A, B, and C in common, but $\mathcal{M}$ consists in addition of the points P_1- P_5, while $\mathcal{M}'$ has the points P_1 - P_5. Although these two configurations appear to be quite different, their city-block distances are equal. This effect is high-lighted by including the lines of the network representation of $\mathcal{M}$ (left panel) and $\mathcal{M}'$ (right panel), and an internal node H that lies in the metric segment of all pairs of points for which the first is selected from the set $\{A, B, C\}$, and the second from either P_1- P_5 or $P_1 - P_5$. It is clear that the only difference between the left panel and the right panel of Figure 5.4 is that they are different embeddings of the same network. Only the part above and to the left of H, the internal node called the *hub*, shows a reflection along the 45^o direction, while the part below and to the right of the hub is the same.

A general formulation of this phenomenon is as follows. Partial isometries occur in city-block spaces whenever the set of objects can be partitioned into subsets $\{F_1, F_2, F_3, \cdots\}$ in such a way that the coordinates of objects from different subsets are either monotonically ascending ($z_{At} \leqslant z_{Bt} \leqslant z_{Ct} \leqslant \cdots$) or monotonically descending ($z_{At} \geqslant z_{Bt} \geqslant z_{Ct} \geqslant \cdots$) for any t, with $A \in F_1$, $B \in F_2$, and $C \in F_3$. This condition implies that we can define a hub in the intersection of all metric segments of points selected from any pair of consecutive subsets. In the network representation, all between-subset distances are thus channeled through (one or more) hub(s). In the embedding of the network in city-block coordinates, we can apply reflections within subsets without altering either the within-subset distances (since reflections do not change distance) or the between-subset distances (since distances to the hub remain unaltered). Summarizing, while the coordinate space is not unique under the monotone subset condition, there is only one network, which merely has different embeddings. Both representations allow an interpretation only in terms of the several within-subset constellations and the global order of the subsets.

5.3 Discrete models that are special cases of the city-block model

Some discrete models of similarity are special cases of the city-block model, and therefore we can make network representations by the same token. One may define these discrete models as structures on subsets of objects, but also as city-block models with binary coordinates. We will first discuss a fundamental property of all discrete models in terms of a condition on subsets, called *lattice betweenness*, and show that lattice betweenness is a special case of metric-segmental betweenness when all coordinates are binary. The most general of all discrete models considered is the distinctive features model, a distance model based on the symmetric set difference, well known to be equivalent to the city-block model on binary coordinates. We then discuss the common features (or additive clustering) model, and show how to obtain

a network representation for this model, too. After a discussion of the conditions for obtaining a perfect solution, for both the common and the distinctive features model, we turn to two special cases, the partitioning model with main effects, and finally the additive tree model.

Lattice betweenness of feature sets

Restle (1959) tried to justify a metric analysis of psychological similarity from set-theoretic considerations, by using the concept of betweenness of sets of qualitative elements and the symmetric set difference as a distance measure[4]. We first discuss the nature of betweenness in this context, returning to the set-theoretic distance in the next section. The common definition in logic for betweenness of sets is to say that if $\mathcal{S} = \{S_1, \cdots, S_i, \cdots, S_m\}$ is a family of subsets of some set of arbitrary or qualitative elements, S_j is between S_i and S_k if the following condition holds:

$$(S_i \cap S_k) \subseteq S_j \subseteq (S_i \cup S_k) \tag{5.8}$$

Thus, to be between S_i and S_k, subset S_j has to share at least all elements common to them, while it cannot have elements not present in either of them. The set of all subsets ordered by the inclusion operator $\subseteq$ is a complete lattice (*cf.* Davey & Priestley, 2002). Therefore, we refer to Equation 5.8 as *lattice betweenness*. To clarify the relation of lattice betweenness and metric-segmental betweenness, we have to make explicit the reliance of the subsets in $\mathcal{S}$ on the base set of qualitative elements. Let this base set be $\mathcal{F} = \{F_1, \cdots, F_t, \cdots, F_T\}$, where the T elements are called *features*[5]. We define the feature matrix $\mathbf{E} = \{e_{it}\}$ as an $m \times T$ binary incidence matrix, where $e_{it} = 1$ if S_i has feature F_t, and $e_{it} = 0$ if not. Thus, the rows of $\mathbf{E}$ characterize an object in terms of a subset of features, while the columns of $\mathbf{E}$ characterize a feature in terms of a subset of objects.

We now show that lattice betweenness is a special case of metric-segmental betweenness. For metric-segmental betweenness between A, B, and C to hold, the coordinates of a city-block configuration have to be either monotonically ascending ($z_{At} \leqslant z_{Bt} \leqslant z_{Ct}$) or monotonically descending ($z_{At} \geqslant z_{Bt} \geqslant z_{Ct}$) for all t (if B is between A and C). Transferring this condition to the binary coordinates in $\mathbf{E}$, we must have either $e_{it} \leqslant e_{jt} \leqslant e_{kt}$ or $e_{it} \geqslant e_{jt} \geqslant e_{kt}$ for all t (if S_j between S_i and S_k). To get from here to Equation 5.8, consider all eight possible (0, 1)-patterns of e_{it}, e_{jt}, and e_{kt}. One may easily verify that six of them satisfy monotonicity, while two of them indicate violation of monotonicity. In particular, violation occurs if

$$(1 - e_{it})e_{jt}(1 - e_{kt}) = 1 \quad or \quad e_{it}(1 - e_{jt})e_{kt} = 1 \tag{5.9}$$

for any t. Interpreting Equation 5.9 in terms of features, we see that metric-segmental betweenness implies that the center subset S_j cannot possess any feature F_t that the

[4]The logician Nelson Goodman already studied the order and topology of qualities in his 1951 book *The Structure of Appearance*, a revised version of his 1940 doctoral thesis *A Study of Qualities* (Harvard University). Galanter (1956) introduced Goodmans ideas in psychology and put them to work with some preliminary experimental findings on color vision.

[5]The index t and parameter T were used earlier for the dimensions of the city-block space, but there is no danger of confusion, as it will turn out that features have exactly the same role as dimensions.

two outer subsets S_i and S_k fail to have, and that S_j also cannot lack any feature F_t that the two outer subsets S_i and S_k both possess. Thus, Equation 5.9 is equivalent to $\overline{S_i} \bigcap S_j \bigcap \overline{S_k} = \emptyset$ and $S_i \bigcap \overline{S_j} \bigcap S_k = \emptyset$ holding at the same time (this is the formulation in Definition 2 used by Restle, 1959), which is in turn equivalent to Equation 5.8. Therefore, a single notion of betweenness for a finite set of points applies equally well in continuous space as in feature space.

Distinctive features model

What metric can we use in a representation of objects as subsets of features? Natural candidates for a distance between subsets are functions of the symmetric set difference,

$$d(S_i, S_j) = \mu\left[(S_i \cup S_j) - (S_i \cap S_j)\right] = \mu\left[(S_i - S_j) \cup (S_j - S_i)\right], \tag{5.10}$$

where $\mu[\cdot]$ is some measure function, usually just a count of the features in the subset[6]. The first part of Equation 5.10 expresses the symmetric set difference in terms of the subset of relevant features (i.e., features in the union) that is not common to the two objects (i.e., features in the intersection). The second part of Equation 5.10 expresses the same notion in terms of the total number of features that belong to S_i but not to S_j (distinctive features for S_i with respect to S_j) and those that belong to S_j but not to S_i (distinctive features for S_j with respect to S_i). Because of the latter formulation, Tversky (1977) has called a model based on equation Equation 5.10 a *distinctive features* model. Note that we do not interpret the term "distinctive" as a qualification of the features (as do Navarro and Lee (2004) in their Modified Contrast Model), but as a qualification of what contributes to the similarity or difference in *pairs of objects*.

One of Restle's (1959) results was that lattice betweenness is equivalent to additivity of the distinctive feature distance, i.e., $d(S_i, S_k) = d(S_i, S_j) + d(S_j, S_k)$. Although in the present context this result readily follows from the equivalence of lattice betweenness and metric-segmental betweenness, it is instructive to derive it explicitly here (via the feature coordinates in $\mathbf{E}$). Suppose $\mu[\cdot]$ is a weighted count measure with weight η_t for feature F_t. As a preliminary step, note that introduction of the feature coordinates allows us to write Equation 5.10 as

$$d(S_i, S_j) = \sum_t \eta_t \left[(1 - e_{jt})e_{it} + (1 - e_{it})e_{jt}\right], \tag{5.11}$$

from which it follows that

$$d(S_i, S_j) = \sum_t \eta_t (e_{it} - e_{jt})^2 = \sum_t |z_{it} - z_{jt}|, \tag{5.12}$$

with $z_{it} = \eta_t e_{it}$. Due to the binary nature of e_{it}, we can replace the squares in Equation 5.12 with absolute values. Thus, the distinctive feature distance is a city-block

[6] Restle (1959) mentions that Hays (1958) used the same distance concept, calling it the "implicational difference", and that he used multidimensional scaling to embed feature distances in Euclidean space

distance, where the points are constrained to lie on the corners of a rectangular (hyper-) block, and where the coordinates on any dimension are limited to two values, zero or η_t. Each feature splits the objects into two classes, and η_t measures how far these classes are apart; for this reason, Heiser (1998) called the feature weight η_t a *discriminability* parameter.

Next, consider three points S_i, S_j, and S_k, and assume betweenness in that order; then we may rewrite the distance between S_i and S_j in Equation 5.11 as

$$d(S_i, S_j) = \sum_t \eta_t \left[e_{it}(1 - e_{jt})(1 - e_{kt}) + (1 - e_{it})e_{jt}e_{kt} \right], \tag{5.13}$$

since if S_i has F_t and S_j has not, then S_k cannot have that feature either, so that $e_{it}(1 - e_{jt}) = e_{it}(1 - e_{jt})(1 - e_{kt})$, while if S_i does not have F_t but S_j does, then S_k must have it too, so that $(1 - e_{it})e_{jt} = (1 - e_{it})e_{jt}e_{kt}$. In other words, Equation 5.13 follows from Equation 5.9. With an analogous expression for $d(S_j, S_k)$, we find

$$\begin{aligned} d(S_i, S_j) + d(S_j, S_k) &= \sum_t \eta_t \left[e_{it}(1 - e_{jt})(1 - e_{kt}) + (1 - e_{it})e_{jt}e_{kt} \right] \\ &+ \sum_t \eta_t \left[e_{it}e_{jt}(1 - e_{kt}) + (1 - e_{it})(1 - e_{jt})e_{kt} \right] \\ &= \sum_t \eta_t \left[e_{it}(1 - e_{kt}) + (1 - e_{it})e_{kt} \right] = d(S_i, S_k). \end{aligned}$$

This equality establishes the result. The implication is that we have metric segments in feature space that are paths along the corners of a (hyper-) block, or equivalently (Flament, 1963, p. 17) as paths in the lattice spanned by the feature sets. Hence, the distinctive features model can be represented as a weighted graph or network, using the same graph construction strategy as the one used for the general city-block model; for discrete models, Heiser (1998) called these representations *feature graphs*. We can also construct internal nodes in the same way. Recall that internal nodes correspond to additional points that are located in one or more metric segments generated by the original (external) points. From condition Equation 5.8, it follows that this rule is equivalent to choosing internal nodes as intersections of feature sets.

Corter and Tversky (1986) provided the first method to fit the distinctive features model, by constructing a so-called extended similarity tree. They used a three-stage procedure: in the first stage, their procedure fits the best additive tree to the data, which limits the features to be either nested or disjoint; in the second stage, it selects additional features to be included in the model, and the third stage the feature weights are estimated for the total set of features. Heiser (1998) used a two-stage alternating least squares method, which just cycles between improvement of the feature structure and improvement of the weight estimates, without the backbone of the additive tree. A third method was recently proposed by Navarro and Lee (2004) as a special case of a more general approach, in which they used maximum likelihood estimation assuming that the similarities are normally distributed with common variance, and employing a greedy heuristic to find the feature sets. These methods were all developed independently, and what their relative merits are, is an open question. There are only a few applications without a priori known feature structure. Parault and Schwanenflugel (2000) used extended similarity trees

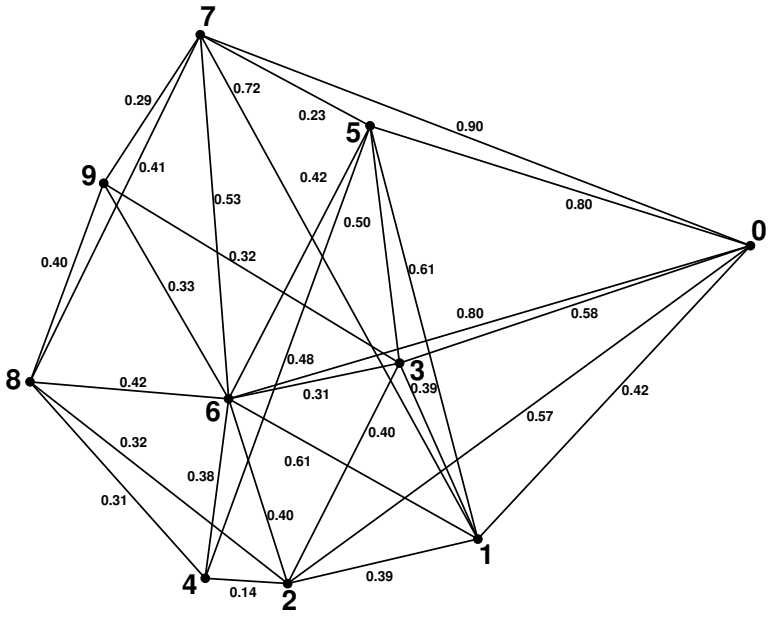

Figure 5.5: Network representation of distinctive features model for the *number* data, without internal nodes. Nodes labeled by stimulus value.

to study the development of childrens categorical knowledge of attention. Heiser and Meulman (1997) used the distinctive features model to cluster profiles of binary multivariate data.

We now demonstrate the construction of a feature network with an example of data collected by Shepard, Kilpatric, and Cunningham (1975), who obtained ratings of similarity between all pairs of integers from zero to nine, considered as abstract concepts. For ease of comparison, we use the same twelve features as Corter and Tversky (1986) found with their EXTREE method. Features in models like these are undefined qualitative elements, but are interpretable by listing the stimuli that have them. The first three form exclusive subsets: the additive and multiplicative identities $F_1 = \{0, 1\}$, powers of two $F_2 = \{2, 4, 8\}$, and a heterogeneous subset of remaining integers $F_3 = \{3, 5, 6, 7, 9\}$. Next, we have the nested features primes larger than three $F_4 = \{5, 7\}$, multiples of three $F_5 = \{3, 6, 9\}$, powers of three $F_6 = \{3, 9\}$, and the first two powers of two $F_7 = \{2, 4\}$. There are five more features that form overlapping subsets: sets of consecutive integers $F_8 = \{0, 1, 2, 3\}$, $F_9 = \{7, 8, 9\}$, $F_{10} = \{0, 1, 2, 3, 4\}$, and $F_{11} = \{4, 5\}$, and the multiples of two, or even numbers $F_{12} = \{2, 4, 6, 8\}$. Finally, we included two unique features $F_{13} = \{0\}$ and $F_{14} = \{1\}$, since otherwise zero and one would have identical feature sets, so that they would not be distinguished in the model, obtaining mutual distance of zero, and would become disconnected from the network.

After estimating the weights using nonnegative least squares (Frank & Heiser, in press a; Heiser, 1998), and applying our basic edge deletion method of dropping the direct edge between two points whenever an intermediate point exist in their metric segment, one gets the network displayed in Figure 5.5. This solution accounts for 98.36% of the dispersion (raw sum of squares) of the data, using 29 edges and 14 parameters. The network itself is a discrete structure in fourteen-dimensional space, but it was embedded in the Euclidean plane by multidimensional scaling of the estimated city-block (feature) distances with the program PROXSCAL (Heiser & Busing, 2004), using a simplex start and allowing a ratio transformation. All edge lengths are included in Figure 5.5 since the Euclidean distances in the plot only approximate them. In the embedded network, the three major features F_1, F_2, and F_3 differentiate well, as do the nested features F_4, F_5, F_6, and F_7. With his large number of connections, stimulus 6 clearly exhibits its overlapping position as a member of the even numbers on the bottom-left and the multiples of three in the center. At the (bottom-)right side of the plot, we have the overlapping features F_8 and F_{10}, and on the top the primes F_4 and top-left the large numbers F_9. Therefore, it appears that the embedded network successfully displays the major characteristics of the distinctive features model in an accessible way.

Nevertheless, the introduction of internal nodes can make the structure even more transparent. The natural choice of internal nodes in the distinctive features model is to identify a cluster in the high-dimensional feature space with a new point that is located in the intersection of the features shared by the objects in that cluster. For example, the internal node corresponding to the cluster $C_1 = \{0, 1\}$ has the features F_1, F_8, and F_{10}, since zero and one share exactly these features. This way of defining internal nodes ensures that the distance between the two members S_i and S_j of cluster C_ℓ splits: $d(S_i, S_j) = d(S_i, C_\ell) + d(C_\ell, S_j)$, because for an additive measure μ we have $d(S_i, S_j) = \mu(S_i - S_j) + \mu(S_j - S_i) = \mu(S_i - C_\ell) + \mu(S_j - C_\ell)$ when $C_\ell = S_i \cap S_j$, and $d(S_i, C_\ell) = \mu(S_i - C_\ell)$ since $\mu(C_\ell - S_i) = 0$. Similarly, we have $d(S_i, C_\ell) = d(S_i, C_k) + d(C_k, C_\ell)$ for members of two nested clusters with $S_i \subset C_k \subset C_\ell$. These additivities lead to better interpretable paths in the network and a low degree for the external nodes, especially if the features are nested or disjoint. Nested features lead to nested clusters, represented as a chain of internal nodes. Let us see how this representation works for the digit data.

The introduction of internal nodes for all clusters corresponding to the 12 features, as well as five extra internal nodes associated to the objects 2, 5, 7, 8, and 9, for which we fitted additional unique features, lead to a network of 27 nodes in 19 binary dimensions in city-block space. Including unique features for the other objects did not improve the fit. We obtained a PROXSCAL embedding with the same options as before; Figure 5.6 displays the result. The seventeen internal nodes are plotted as open dots, while the ten object nodes (external nodes or leaves) are plotted as solid dots. Every object node with a unique feature is connected to the rest of the network via an (unlabeled) internal node with a spike of length equal to the unique feature weight. Note that all paths from 0 and 1 go through {0, 1} and then through {0, 1, 2, 3}, all paths from 2 go through {2, 4} and {0, 1, 2, 3}, all paths from 3 go through {3, 9} and {0, 1, 2, 3}, all paths from 4 go through {2, 4} and {4, 5}, and so on. In other words, in this example all objects have only two direct neighbors, which are always

internal nodes (clusters), except for 0 and 1, which have only one direct neighbor because they differ only in their unique features, and 7 and 9, which have a mutual link in addition to their two cluster connections.

Figure 5.6 also shows how nesting of features leads to nested clusters in a chain of internal nodes. The most important chains are: the small numbers {(0, 1), (0, 1, 2, 3), (0, 1, 2, 3, 4)}, the even numbers {(2, 4), (2, 4, 8), (2, 4, 6, 8)}, and the prime numbers plus powers and multiples of three {(3, 9), (3, 6, 9), (3, 5, 6, 7, 9)}. The three chains are connected in a triangle in the center of the display, which forms a complete sub-network of internal nodes together with the medium-sized numbers (4, 5) and the large numbers (7, 8, 9). All object nodes are connected via one or two paths to this basic complete sub-network. The path can be linked either directly, like from 6 to (2, 4, 6, 8) and from 8 to (7, 8, 9), or go through the closest node in one of the chains, like from 2 to (2, 4) in the chain of even numbers, or from 2 to (0, 1, 2, 3) in the chain of small numbers. The global structure of the embedding appears to consist of a basic plane with the powers of two (2, 4, 8) on one side and the powers of three (3 and 9) on the other side, with small numbers at the right and large numbers at the left. It appears that objects 5, 6, and 7 do not fit well into this plane, either because they have a large unicity (5, 7) or because they share only partly features from both sides (6 is a multiple of both two and three, but not a power of them). The identities 0 and 1 have an eccentric position with large unicity, but as a cluster, they are close to the small numbers.

Additive clustering or the common features model

Shepard and Arabie (1979) proposed an additive clustering model, which builds up the similarity $s(S_i, S_j)$ between S_i and S_j from unrestricted binary features, according to the rule

$$s_{ij} = s(S_i, S_j) = \mu\left[S_i \cap S_j\right] = \sum_t \theta_t e_{it} e_{jt}, \tag{5.14}$$

where the θ_t are again nonnegative weight parameters. This model thus uses a weighted count of the features in the intersection of the feature sets of each object in a pair. Since the model only takes features into account that the pair of objects have in common, Tversky (1977) has called it a *common features model*. Mirkin (1987) developed the model independently under the name *qualitative factor analysis*, around the same time as Shepard and Arabie, and adjusted it to the analysis of contingency tables (the two-mode case) in Mirkin (1996). Arabie and Carroll (1980) and Carroll and Arabie (1983) developed algorithms for finding the feature sets and the feature parameters of the additive clustering model and its three-way generalization. Soli, Arabie, and Carroll (1986) reported an application of the three-way additive clustering model. More recent algorithmic strategies are given in Mirkin (1990, 1998), Chaturvedi and Carroll (1994), and Ten Berge and Kiers (2005), among others.

In the additive clustering model, each feature defines a cluster of objects. The unrestricted nature of the features implies that the clusters need not be exclusive and may overlap. As noted by Carroll and Corter (1995), graphical representations of non-nested overlapping clustering are usually complex and difficult to in-

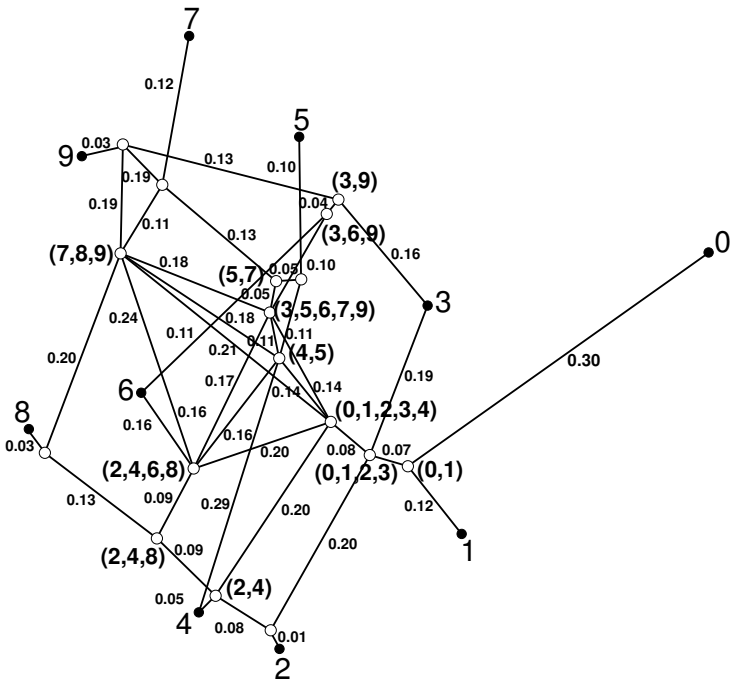

Figure 5.6: Network representation of distinctive features model for the number data, with internal nodes. Solid dots are stimuli labeled by stimulus value, open dots are internal nodes labeled by subset.

terpret. Shepard and Arabie (1979) used a two-dimensional projection of a three-dimensional city-block embedding of the original data, and then added contour lines around sets of points that correspond to clusters in the additive clustering solution. Carroll and Pruzansky (1980) proposed representing non-nested clustering by multiple trees, and (Corter & Tversky, 1986) by extended trees. What we want to show now is that the clusters derived from the additive clustering model have a natural representation as a feature network. To demonstrate this possibility, we express the common features model as a special case of the distinctive features model. This relationship was first established by Sattath and Tversky (1987).

Suppose that we have a feature set $\mathcal{F}$, coded in a feature matrix $\mathbf{E}$, and weight parameters $(\hat{\theta}_1, \cdots, \hat{\theta}_t, \cdots, \hat{\theta}_T)$, that approximate some similarity ς_{ij} according to the common features model (Equation 5.14), where we denote the approximation by $\hat{s}_{ij} = \sum_t \hat{\theta}_t e_{ij} e_{jt}$. We want to demonstrate that it is possible to form a specific linear transformation $\hat{d}_{ij} = 2K - 2\hat{s}_{ij}$ that follows exactly a distinctive features model, where K is some constant that we will specify later. We can use the same feature set $\mathcal{F}$, but we have to append to it a set of m unique features. A *unique feature* is a feature with only one object associated to it, with non-negative weight. To distinguish the

features in **E** from the unique features, we call the former *shared features*, since there are always two or more objects sharing a non-unique feature. The feature matrix of a set of unique features is diagonal, so that by themselves they form an additively decomposable metric associated with a star graph, as we saw in the discussion of Figure 5.2. We will use the notation e_{it^*} for the unique features, with $t^* = 1, \cdots, m$, and with the understanding that $e_{it^*} = 1$ if $i = t^*$ and $e_{it^*} = 0$ otherwise. Without danger of confusion, we use $\alpha_i = \sum_{t^*} \alpha_{t^*} e_{it^*}$ for the unique weight of object i.

To let the switch from common features model to distinctive features model work, it suffices to take identical weights for the shared features, while the weights for the unique features are a simple function of the shared feature weights, specified as follows:

$$\begin{aligned} \hat{\eta}_t &= \hat{\theta}_t \\ \hat{\alpha}_i &= K - \sum_t \hat{\theta}_t e_{it}. \end{aligned} \tag{5.15}$$

The constant K can be chosen freely as long as it does not make the weights α_i negative, i.e., as long as it satisfies $K \geqslant \max_i \sum_t \hat{\eta}_t e_{it}$. From Equations 5.15 we have $K = \hat{\alpha}_i + \sum_t \hat{\eta} e_{it}$, so that we may write

$$\begin{aligned} \hat{d}_{ij} &= 2K - 2\hat{s}_{ij} = K + K - 2\sum_t \hat{\theta}_t e_{it} e_{jt} \\ &= \hat{\alpha}_i + \sum_t \hat{\eta}_t e_{it} + \hat{\alpha}_j + \sum_t \hat{\eta}_t e_{jt} - 2\sum_t \hat{\eta}_t e_{it} e_{jt} \\ &= \left[\sum_t \hat{\eta}_t e_{it} + \sum_t \hat{\eta}_t e_{jt} - 2\sum_t \hat{\eta}_t e_{it} e_{jt}\right] + \left[\hat{\alpha}_i + \hat{\alpha}_j\right] \\ &= \sum_t \hat{\eta}_t |e_{it} - e_{jt}| + \sum_{t^*} \hat{\alpha}_{t^*} |e_{it^*} - e_{jt^*}|. \end{aligned} \tag{5.16}$$

Note that whenever we have the approximation $\hat{d}_{ij}$, we also recover $\hat{s}_{ij} = K - \frac{1}{2}\hat{d}_{ij}$. Also, note that $(\eta_1, \cdots, \eta_t, \cdots, \eta_T)$ and $(\alpha_1, \cdots, \alpha_{t^*}, \cdots, \alpha_m)$ should not be seen as a set of $T + m$ independent parameters, because both are functions of the T parameters $(\theta_1, \cdots, \theta_t, \cdots, \theta_T)$. Clearly, $\hat{d}_{ij}$ in Equation 5.16 has the desired form of a distinctive feature distance, since the sum of two feature distances is again a feature distance with dimensionality equal to the sum of the two original dimensionalities. Therefore, we can check all triads of points for lattice betweenness to see which edges of the network we can delete, as usual.

It is often useful in this approach to the graphical representation of the common features model to define m internal nodes, one for each object, with shared features that are the same, but without unique features. The effect will be that the feature graph displays the structure of the shared features in its internal nodes, each of which corresponds to (and can be labeled with) exactly one object. In addition, each internal node has one unique edge (a spoke toward one external node) attached to it, the length of which indicates the relative distance of an object towards all others. This spoke (and its length) is analogous to a unique factor (and its variance) in factor analysis. Hence, Mirkin (1987) name qualitative factor analysis for the common features model is well chosen.

In factor analysis, the diagonal of the correlation matrix to which we fit the model is constant, and the unique factors are necessary to account for the variance left unexplained by the common factors. In the common features model, the shared features produce diagonal terms equal to $\hat{s}_{ii} = \sum_t \hat{\theta}_t e_{it}$, the sum of the weights that an object possesses, and these will generally not be constant either. Hence, if one would like to account for the diagonal elements of the similarity matrix, one would need to append a set of m unique features to the common features model as well, that is, write the model as

$$\underline{s}(S_i, S_j) = \sum_t \theta_t e_{it} e_{jt} + \sum_{t^*} \beta_{t^*} e_{it^*} e_{jt^*},$$

with e_{it^*} denoting the unique feature of object i and $\beta_{t^*} = \beta_i \geqslant 0$ its non-negative weight. The diagonal elements are equal to the sum of all weights relevant for object i:

$$\underline{s}(S_i, S_i) = \sum_t \theta_t e_{it} + \sum_{t^*} \beta_{t^*} e_{it^*} = \sum_t \theta_t e_{it} + \beta_i,$$

while the off-diagonal elements remain the same as before, since $\sum_{t^*} \beta_{t^*} e_{it^*} e_{jt^*} = 0$ if $i \neq j$. Therefore, if we want diagonal elements equal to some constant value, that is, $\underline{s}(S_i, S_i) = K$, the unique weights for the extended common features model should be chosen as

$$\hat{\beta}_i = K - \sum_t \hat{\theta}_t e_{it},$$

that is, identical to the unique weights under the extended distinctive features model in Equation 5.15, because $\hat{\eta}_t = \hat{\theta}_t$. Since in practice one usually does not model the diagonal elements of the similarity matrix, the issue never seems to arise. However, to make the model complete, the common features model needs the same unique features as the distinctive features model.

In the distinctive features model, the effect of the unique features with weights $\hat{\alpha}_i$ defined in Equation 5.15 also is to make the sum of the weights for each object constant. This property can be expressed geometrically by calculating the feature distance of object S_i with respect to the origin O (internal node with all-zero profile), and inserting Equation 5.15:

$$d(S_i, O) = \sum_t \hat{\eta}_t e_{it} + \hat{\alpha}_i = \sum_t \hat{\eta}_t e_{it} + K - \sum_t \hat{\eta}_t e_{it} = K.$$

Hence, the common features model for similarity matrices with equal self-similarities is a special case of the distinctive features model: if the unique feature weights satisfy Equation 5.15, then reversing the argument in Equation 5.16 shows that the distances satisfy a common features model. In practical terms, for any fitted common features model we can find an equally well fitting distinctive features model, with object nodes at constant distance from the origin, with the same shared features and feature weights, and with the same number of independent parameters.

As an example of the network representation of the common features model, consider the body parts data collected by Miller (1969), which was reanalyzed by

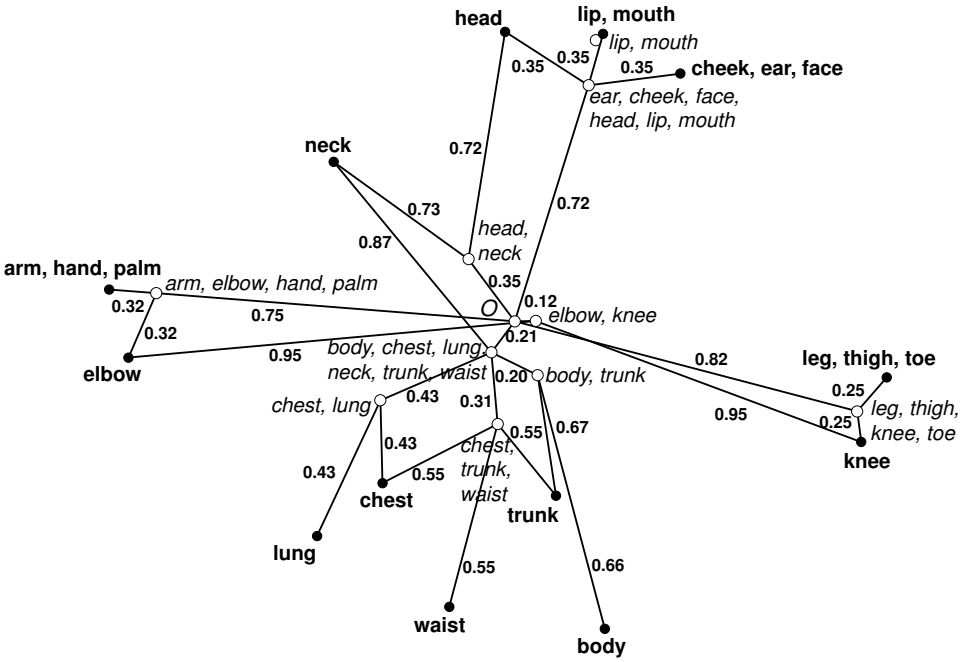

Figure 5.7: Network representation of common features model for *body-parts* data, with internal nodes.

Carroll and Chang (1973), and Shepard and Arabie (1979). Miller's data on the perceived similarity of 20 body parts are counts of the number of times in which 50 subjects, in a free sorting task, put a pair of stimuli into the same group. As noted by Shepard and Arabie, the body parts had been chosen based on a rather clear hierarchy of anatomical inclusion, but with some ambiguities. We have used the same 10 common features and weights found by their ADCLUS procedure, which accounted for 95.6% of the variance in the similarities. Using Equation 5.15 to calculate weights for the shared and unique features under the distinctive features model and applying the general graph construction rule we obtain the network representation in Figure 5.7. In this representation, eleven internal nodes have been included to reduce the degree of some of the nodes. One of them is labeled by O, and can be interpreted as the root or the origin of the network, since it is defined by a profile of zeros on all features. The other internal nodes are labeled by the subsets found by Shepard and Arabie. They are defined by the intersection of the features of the objects in the subset that they represent. The network clearly shows that there are four major clusters: a trunk cluster (consisting of body, chest, lung, neck, trunk, and waist), a leg cluster (knee, leg, thigh, toe), arm cluster (arm, elbow, hand, palm), and a head cluster (ear, cheek, face, head, lip, mouth), which is consonant with previous

analyses. In this solution, we do not find evidence that trunk, leg, arm, and head are especially close to their closest cluster point, to warrant the higher-order status that they had in the Carroll and Chang (1973) solution.

A strong point of the current representation is that violations of hierarchical structure are recognizable as cycles in the network. One major cycle is between the origin, the trunk cluster and the head cluster (via head, neck), and another one is between the origin, the arm cluster, and the leg cluster (via elbow, knee). These cycles arise from the presence of feature 5, which connects two physically neighboring parts, head and neck, and feature 10, which connects two functionally analogous parts, elbow and knee. Thus, in addition to its simple portrayal of the additive clustering solution, in which the clusters themselves can be included in a natural way, the network representation of a common features model allows an immediate diagnosis of departures from purely hierarchical structure. Note the following regularities in Figure 5.7. The sum of the line lengths from each leave node (solid dot) to the origin is constant and equal to 1.07 (up to rounding error). One can read off the dissimilarity between two leaves as the length of their shortest connecting path. At the same time, one can read off their similarity as the sum of the line lengths of a path from the origin to the smallest cluster that they share.

Exact fit of feature models

Is there a feature set that always yields an exact fit to an arbitrary (dis)similarity matrix under these feature models? There is no exact answer in the literature to this question, but the previous section shows how to obtain one. It is clear that under the common features model a basis consisting of all size-two clusters corresponding to all pairs of objects would be sufficient to fit any similarity matrix exactly. Let us denote the features of this basis by $e_{i(k,l)}$, where k, l varies over all ordered pairs, and we have the property $e_{i(k,l)} = 1$ if $i = k$ or $i = l$, and $e_{i(k,l)} = 0$ otherwise. Since $e_{i(k,l)}e_{j(k,l)} = 0$ for all k, l except if $i = k$ and $j = l$, there is exactly one feature for each similarity, so that we can choose $\hat{\theta}_{(k,l)} = \varsigma_{kl}$, obtaining an exact fit $\varsigma_{ij} = \hat{s}_{ij}$.

Under the distinctive features model, we can use the same basis of all size-two clusters, but we need again to include unique features to reproduce any dissimilarity matrix up to a known additive constant. It is not hard to show that in this feature structure no object is between any other object, so that the feature network is a complete graph[7]. The specification of the parameters is

$$\begin{aligned} \hat{\eta}_{(k,l)} &= L - \frac{1}{2}\delta_{kl}, \\ \hat{\alpha}_i &= \frac{1}{2}\sum_{j \neq i}\delta_{ij} - (m-2)L, \end{aligned} \tag{5.17}$$

[7]For three objects A, B and C, the relevant features are AB, AC, and BC. Thus, it suffices to consider $A = \{AB, AC\}, B = \{AB, BC\}$, and $C = \{AC, BC\}$. Whatever object is chosen as the middle one, it violates the requirement defined in Equation 5.9 that it should not lack any feature that the two outer objects possess. For instance, A and C share AC, but B lacks it.

where L is some positive constant. With these weights, the feature distance becomes

$$\begin{aligned} d(S_i, S_j) &= \sum_{k,l} \hat{\eta}_{(k,l)} |e_{i(k,l)} - e_{j(k,l)}| + \hat{\alpha}_i + \hat{\alpha}_j \\ &= 2(m-1)L - \frac{1}{2}\sum_{j \neq i} \delta_{ij} - \frac{1}{2}\sum_{i \neq j} \delta_{ij} + \delta_{ij} - 2L \\ &\quad + \frac{1}{2}\sum_{j \neq i} \delta_{ij} + \frac{1}{2}\sum_{i \neq j} \delta_{ij} - 2(m-2)L \\ &= \delta_{ij}. \end{aligned} \tag{5.18}$$

Thus, for any choice of L in Equation 5.17, we have perfect reconstruction of the dissimilarities. It turns out that adding a constant to the weights of the shared features can be compensated by subtracting (another) constant from the unique features. This indeterminacy is caused by the fact that all pairs of objects differ on the same number of shared features $(m-1)$, and on the same number of unique features (two). We can identify a solution by selecting L so that the smallest unicity becomes zero, for example. However, there is another consideration. When choosing L too small we obtain one or more negative weights $\hat{\eta}_{(k,l)}$ for the shared features, and when choosing L too large we obtain one or more negative weights $\hat{\alpha}_i$ for the unique features, both of which are violations of the model assumptions. Requiring nonnegativity of the two sets of weights in Equation 5.17 gives the following bounds for L:

$$\max_{(i,j)} \delta_{ij} \leqslant L \leqslant \min_i \frac{1}{m-2} \sum_{j \neq i} \delta_{ij}. \tag{5.19}$$

Therefore, we can identify a solution whenever the dissimilarities allow finding an L in the interval Equation 5.19. If no such L exists, we can add the smallest positive constant to the dissimilarities ensuring that Equation 5.19 becomes satisfied. Finding such an additive constant is possible, because the lower bound involves only one dissimilarity, while the upper bound involves the sum of $m-1$ dissimilarities divided by $m-2$, so that the upper bound grows faster than the lower bound. In conclusion, a feature network based on size-two clusters and singletons can always reproduce an arbitrary dissimilarity matrix.

Even though perfect reproduction involves as many as $\frac{1}{2}m(m+1)$ features, while there are merely $\frac{1}{2}m(m-1)$ independent data values, it should be noted that each α-weight can be written as a linear function of the data values, so that we actually do rely on exactly $\frac{1}{2}m(m-1)$ independent quantities. A calculation similar to Equation 5.18 shows that the feature distance of any object to the origin is constant; in particular, we have $d(S_i, O) = L$. Since the square root of the feature distance is Euclidean (see Equation 5.12), it follows that the vertices of the complete graph that perfectly reproduces an arbitrary dissimilarity matrix are located on a hypersphere of dimension $\frac{1}{2}m(m+1)$ with radius $\sqrt{L}$.

Partitioning in clusters with unicities: the double star tree

Consider the situation in which the model consists of a set of unique features and a set of shared features, where the latter has the special property of forming a partition of the set of objects. Thus, each shared feature is disjoint from (or non-overlapping with) any other shared feature, and no object lacks a shared feature (in addition to its unique feature). Without the presence of unique features, this case would be a standard clustering task for which several methods have been developed (*cf.* Hubert, Arabie, & Meulman, 2001). As we saw earlier, unique features can be represented as a star graph. A partitioning in T subsets can be represented as a star graph, too, in which each subset is a vertex and the center of the star is again the origin (an internal node with zero on all features). Fitting a model that is the sum of two star graphs is a simple special case of Carrolls (1976) multiple tree structure approach, but surprisingly no one has considered it in any detail[8].

The graphical representation obtained for the sum of the distances in the star graph of the unique features and the distances in the star graph of the partitioning has a particularly simple form. We need one internal node for the origin, and T other internal nodes (where T is the number of clusters), each having a single non-zero value for one of the features defining the partitioning. With our usual graph construction procedure of eliminating direct lines when two nodes are reachable through another node in their metric segment, we obtain a graph in which each internal cluster node connects only with the origin and with the leaves that constitute the cluster. Thus, the origin node has degree T, the cluster nodes have degree $n_t + 1$, where n_t is the number of objects in cluster t, and the object nodes are leaves with degree one. Any distance between two objects in different clusters equals the sum of four line lengths along the unique path connecting them. Starting with S_i, we have the line from the leave of S_i to the node of the cluster where S_i belongs to, the line from that cluster node to the origin, the line from the origin to the cluster node of S_j, and finally the line from the cluster node of S_j to the leave of S_j. The distance between two objects in the same cluster is just the sum of two line lengths. The graph of the double star tree is simple because it contains no cycles and has only $T + m$ lines. There is also a one-to-one relation between line lengths and feature weights.

For the Shepard et al. (1975) number data, analyzed earlier with the general distinctive features model and displayed in Figures 5.5 and 5.6, Hubert et al. (2001) repeatedly found the optimal partition {(0, 1), (2, 4, 8), (3, 6, 9), (5, 7)}, with different clustering criteria. Therefore, we adopted this partitioning and estimated weight parameters for the shared and unique features with nonnegative least squares. Figure 5.8 displays the resulting network. We see that the graph has all the properties described in the previous paragraph. It has no cycles, and since $m = 10$ and $T = 4$ in this case, it contains 4 + 10 = 14 lines. The origin only connects with the four cluster nodes, and each object only with the cluster node of its own cluster. The distance between 1 and 5, which belong to different clusters, is the sum of the four line

[8]The closest example of a partitioning model with unicities that we could find in the literature is one of the hierarchical tree structure models proposed by Carroll and Chang (1973), which they call the "branches only" model. The partitioning occurs incidentally in their example of the body-parts data, because the fitted tree is not fully resolved

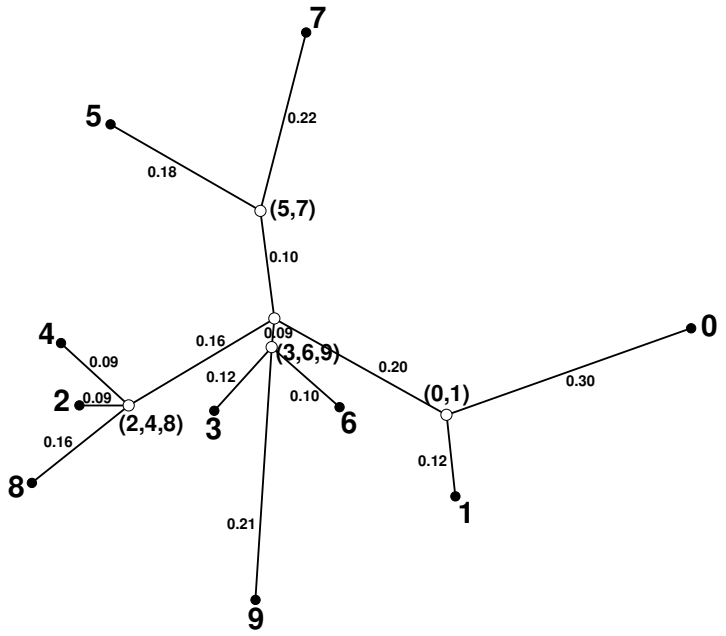

Figure 5.8: Network representation of double star tree for the *number* data.

lengths along the path $(1) - (0,1) - (O) - (5,7) - (5)$, amounting to 0.12 + 0.20 + 0.10 + 0.18 = 0.60 (compared to 0.61 in Figure 5.5). The distance between 2 and 8, which belong to the same cluster, is the sum of the two line lengths along the path $(2) - (2,4,8) - (8)$, amounting to 0.09 + 0.16 = 0.25 (compared to 0.32 in Figure 5.5). The double star tree accounts for 94.84% of the dispersion (compared to 98.36% for the more general model), so it still has a good fit. It is easy to check that none of the within-cluster distances is larger than any between-cluster distance, which is a sign of the quality of the partitioning; for example, compare the largest distance of 0.42 within cluster (0, 1) with a smallest distance of 0.44 between 4 and 6 in the powers of two and the multiples of three clusters. It is a strong point of the double star tree that it models within-cluster distances in addition to between-cluster distances. In contrast, other partitioning methods usually assume that the within-cluster distances are random or zero.

Additive tree model

Consider building up a model with one feature defining a partitioning in two clusters and a full set of unicities, and introduce an extension with features that are limited to be proper subsets of previous clusters (excluding singletons, since there is no need to duplicate the unicities). This construction leads to at most $m - 3$ shared

features that are either nested or disjoint. In terms of feature sets, the implication is that any three objects can be labeled so that $S_i \cap S_j = S_i \cap S_k \subset S_j \cap S_k$. Now the feature distance satisfies a special property that is characteristic for an *additive* or *weighted tree*, called the additive inequality or the four-point condition (Buneman, 1971; 1974). A tree is a connected graph without cycles, and the qualifier additive underlines the property that the distance between any two nodes in a weighted tree is the sum of the weights (line lengths) along the shortest path connecting the nodes. Tversky (1977) was the first to give an interpretation of an additive tree in terms of the distinctive features model, and advocated its use as a practical simplification of his more general Contrast Model. Colonius and Schulze (1981) gave a measurement-theoretical characterization of the tree structure in terms of topological relations between pairs of objects and described corresponding sorting tasks for data collection.

Cunningham (1974, 1978), Carroll (1976) and Sattath and Tversky (1977) motivated their work on the additive tree by pointing out limitations of the more common hierarchical tree and multidimensional scaling representations as models for similarity data. Given two disjoint clusters in a hierarchical tree, for example, all within-cluster distances are smaller than all between-cluster distances, which are all equal. Such severe constraints do not necessarily hold in an additive tree. Pruzansky, Tversky, and Carroll (1982) offered guidelines for deciding between spatial and tree representations on the basis of data properties such as skewness of the (dis)similarity distribution (under an additive tree model the distance distribution is skewed to the left, and under a spatial model distances are skewed to the right). Carroll, Clark, and DeSarbo (1984) proposed extensions of additive tree model to three-way data. Despite its elegance and flexibility, applications of additive trees in psychology are sparse, except perhaps in categorization research. An example is the study of contrast categories in predicting typicality ratings by Verbeemen, Vanoverberghe, Storms, and Ruts (2001).

Several algorithms are available for fitting an additive tree (see Barthélemy & Guénoche, 1991). The major ones are ADDTREE (Sattath & Tversky, 1977), ADDTREE/P (Corter, 1982), an improved implementation of the ADDTREE algorithm because it allows for using metric information, the closely related and widely used neighbor-joining (NJ) method (Saitou & Nei, 1987), and a least squares method due to De Soete (1983). GTREE (Corter, 1998) uses only metric information to select the nearest neighbor for each object and therefore represents an entirely distinct algorithm from ADDTREE and ADDTREE/P. Viewed as a distinctive features model, the tree is characterized by at most $m - 3$ shared features that are either nested or disjoint, and m unique features. Given the tree structure, we can find anyone of the features by cutting any branch of the tree, causing the objects to fall apart in two exclusive subsets. Repeated cutting of all $2m - 3$ branches gives the complete set of features. Given the feature structure, the tree can be found by the present graph construction method, where each of the $m - 3$ shared features is included as an additional internal node (defined as the intersection of the profiles of the objects sharing the feature). The origin should be included as well; this internal node corresponds to the complement of the subset defined by the first feature. There is a one-to-one relation between line lengths and feature weights. An interesting special case arises if we constrain each internal node to be equal to one of the objects (the "branches only"

model of Carroll & Chang, 1973), which amounts to setting the weight of some of the unique features equal to zero. This constrained model has only $m - 1$ parameters.

Corter and Tversky (1986) found an additive tree for the Shepard et al. (1975) number data with ADDTREE. Using the procedure just outlined, we recovered seven shared features. The weight parameters for the shared and unique features have been re-estimated with nonnegative least squares. Our usual graph construction method yielded the network displayed in Figure 5.9, which is comparable with our earlier results for the more general distinctive features model in Figure 5.6 and the more restricted double star tree model in Figure 5.8. The %DAF of this solution of 95.37 is between those of the other two. There are clear common elements between the three solutions, in particular the fact that they share the clusters (0, 1), (5, 7), (2, 4, 8), and (3, 6, 9). The additive tree refines (2, 4, 8) into (2, 4) versus (8), and (3, 6, 9) into (3, 9) versus (6), while it introduces the super-ordinate class (3, 5, 6, 7, 9) by joining (3, 6, 9) and (5, 7). Remarkable differences between the three solutions are the following. In the general distinctive features model, 9 is close to 7, but not in the two other models; this is due to the cluster of large numbers (7, 8, 9), which joins elements from three of the four major clusters apparent in the other two models. Similarly, the cluster of small numbers (0, 1, 2, 3 ,4) in the general feature model forms a major violation of the hierarchical structure, since it also combines elements from three of the four main clusters in the other two models. Both the tree and the general model join (5, 7) with (3, 6, 9) to form (3, 5, 6, 7, 9), but this cluster does not occur in the partitioning model. In the tree, (2, 4) joins with 8 into (2, 4, 8), but does not continue with (2, 4, 6, 8) like in the general model, since 6 is located in another branch of the tree. All differences are understandable from the structural properties of the three models.

5.4 Discussion

Additivity across dimensions and uniqueness of coordinate system have always been the two most appealing properties of the city-block distance, ever since Landahl (1945) started thinking of models for similarity and difference, and Attneave (1950) started experimenting with them (Arabie, 1991). Undoubtedly, the simplest rule for the combination of psychological differences on different dimensions is to add them up with equal weights (Cross, 1965). This paper has shown that the city-block distance is not only additive across its component dimensions, but also across sequences of intermediate points along certain trajectories in space. As an unexpected consequence, the extra additivity of distance allows dropping the whole coordinate system. If we can embed dissimilarities in a city-block coordinate system, we can equally well embed them in a network.

Our construction of the network representation rested upon the notion of the metric segment between any pair of points in space. A metric segment is the area of all intermediary points for which additivity of distance applies. City-block space has metric segments that are rectangles in two dimensions, cuboids in three dimensions, or hyper-cuboids in more than three dimensions. Since these areas are large enough to accommodate a considerable number of intermediate points in any finite set of

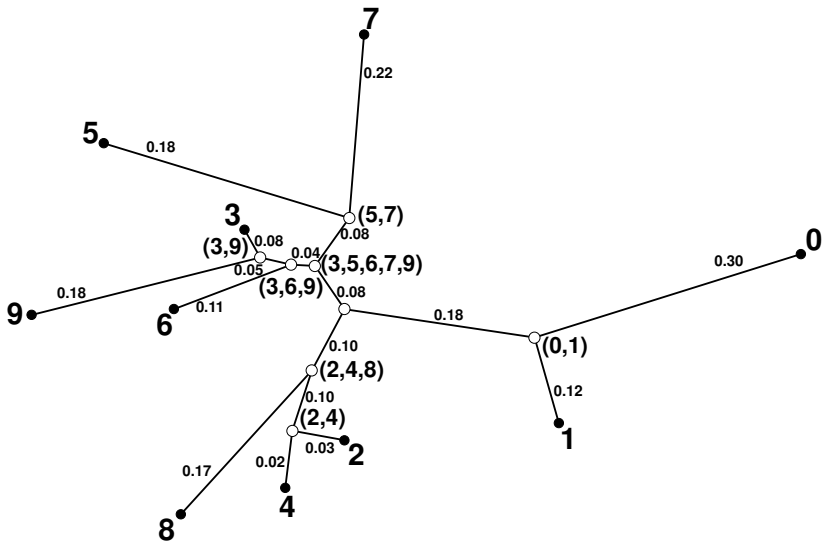

Figure 5.9: Network representation of additive tree for the *number* data.

objects, the possibility of network construction is realistic for city-block models. By contrast, in Euclidean space metric segments are always line segments, and chances of finding intermediate points on line segments are negligible with fallible data on a finite set of objects. We also introduced the general concept of an internal node, which is a supplementary point in the intersection of several metric segments. Internal nodes can be helpful in reducing the complexity of the network, and in making the representation more transparent and better susceptible for interpretation.

A network is coordinate-free, that is, it is entirely determined by the presence or absence of edges between the nodes, and the lengths of these edges; in other words, it exists independently from an embedding in some coordinate system. In some applications, such as the example of the Borg and Leutner (1983) data, one could consider that property undesirable, since coordinates of objects are essential: they are the psychological part of the psychophysical function. Nevertheless, when fitting the city-block model without restrictions enforcing that the dimensions are indeed simple functions of the independent variables, a procedure often used, there is no guarantee whatsoever that the coordinates satisfy the expectations. Indeed, they often do not correspond exactly with the predicted dimensions, as was also clearly the case in our analysis of the Borg and Leutner data in Figure 5.1. In those situations, the coordinate-free representation with internal nodes can be useful in that it offers suggestions of the type of violations that occurred, as we have seen in the discussion of Figure 5.3. For a real test of inter-dimensional additivity, it might still be the best to follow simply Attneave (1950), who predicted observed differences between stimuli varying on two dimensions from observed differences

between stimuli varying only within dimensions and fitted a regression equation.

Coordinate-free models rely merely on distance and local relations. One may argue that these two elements are enough to navigate mentally through cognitive space. There is growing evidence that human navigation in physical space has two distinct means of keeping track of position and orientation during travel: landmark-based navigation and path integration (Klatzky, Beall, Loomis, Golledge, & Philbeck, 1999). While landmark-based navigation depends on some coordinate system - be it Cartesian or with polar coordinates - path integration is a mechanism that builds up a mental image of the trajectory traversed by encoding distances and turns, on the basis of sensed self-velocity, self-acceleration, and self-rotation. Thus, in some circumstances a network representation might have more psychological reality than a coordinate representation, which also often assumes more continuity in psychological space than is warranted by the data.

An important difference between networks for continuous city-block models (most often of low dimensionality) and networks for discrete city-block models (most often of high dimensionality) is the type of embedding needed to achieve an interpretable display. In the first case, the coordinates of the continuous solution often suffice, and no extra embedding is necessary (except for high-dimensional solutions). In the second case, we do need a form of multidimensional scaling for visualizing the nodes and the edges, which adds some arbitrariness to the final display, since several variations in analysis options are possible (type of fit function used, type of possible distance transformation specified, type of start configuration used, and so on). Nevertheless, the linking structure and the edge lengths are invariant. Therefore, when reporting a network, either the edge lengths or the feature parameters themselves should always be included. In addition, the goodness-of-fit between data and reconstructed network distance (network fit) is a more important consideration than the goodness-of-fit between reconstructed network distance and the distances in the visual display (embedding fit).

Network representation of feature structures offers a fruitful framework for theoretical comparison and practical use of a whole range of scaling and clustering methods. For example, our derivation of the common features model as a special case of the distinctive features model is a new result, owing to a more transparant notation than the one used in Sattath and Tversky (1987) and Carroll and Corter (1995). Since our network construction rule applies to continuous and discrete models alike, it turns out to be a unifying factor for understanding the relations between them. Figure 10 gives an overview of these relations. From top to bottom, Figure 10 has six levels, each adding some extra restriction to the model. One-step down from the most general continuous case, the distinctive features model arises from the restriction that coordinate values be binary (where the distance between the two values is not necessarily equal for all dimensions). At the same level of generality, we have Corter and Tversky (1986) extended similarity tree, which is an equivalent form, provided that we allow the tree being unresolved (for instance, if all features overlap without nesting, we can only have an extended similarity tree representation by reducing the tree to a bipartition). Continuing further down in Figure 10, we have:

- *Third level* (*common features model, additive tree*). The additive tree arises from the distinctive features model by the restriction that all features are either nested or disjoint and from the extended similarity tree by the exclusion of marked segments. As shown in this paper, the common features model arises from a restriction on the weights of the unique features, so that the total sum of all

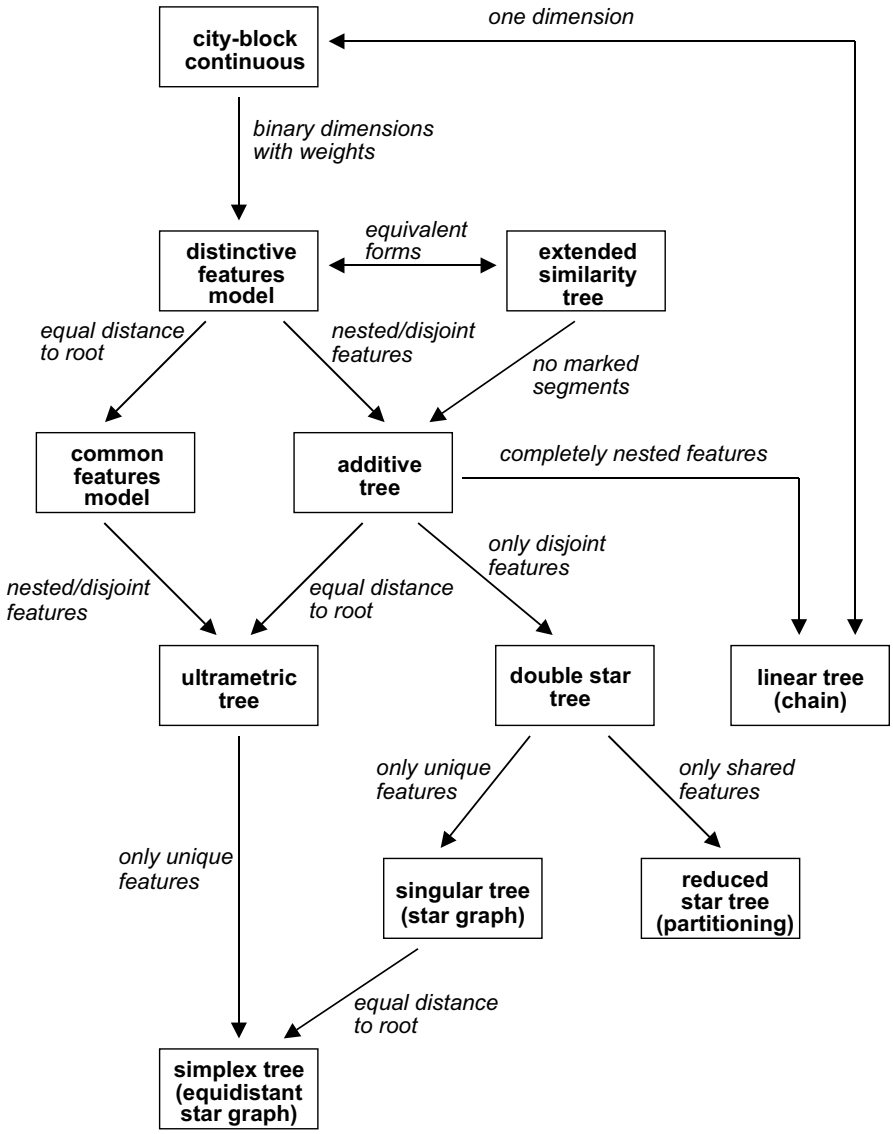

Figure 5.10: Relationships between city-block models.

feature weights is constant. Although Sattath and Tversky (1987) have stated that the distinctive features model and the common features model have the same level of generality, we believe that their argument runs into a contradiction with respect to the diagonal entries of the similarity matrix (see Heiser and Frank (2005), for a more detailed argumentation).

- *Fourth level* (*ultrametric tree, double star tree, linear tree*). The ultrametric tree arises from the additive tree by the restriction that all nodes are equidistant from the root (Carroll, 1976), but it is also a special case of the common features model in which all features are restricted to be either disjoint of nested (Carroll & Corter, 1995). If all features are completely nested and unique feature weights are zero except for one, we have a chain or linear tree (Sattath & Tversky, 1977), which is equivalent to a one-dimensional continuous distance model. As shown in this paper, if all shared features are disjoint, we have a double star tree.

- *Fifth level* (singular tree, reduced star tree). If all unique features in the double star tree are restricted to have zero weights, we obtain a reduced star tree, or simply a partitioning. If all shared features in the double star tree are restricted to have zero weights, we obtain a star graph (Carroll, 1976), also called a singular tree (Sattath & Tversky, 1977).

- *Last level* (*simplex tree*). If the leaves of a star graph are equidistant to the root, that is, if all unique feature weights are equal, we obtain the equidistant star graph, or simplex tree. Equal distances also arise if an ultrametric tree is completely unresolved (that is, the weights of all shared features reduce to zero).

It appears that all known discrete models of similarity fit well into this scheme. They are all special cases of the distinctive features model, and the general rule proposed in this paper produces their usual graphical representations, thanks to the introduction of internal nodes.

One model not mentioned in Figure 5.10, Tversky's (1977) Contrast Model, is decomposable into a symmetric and a skew-symmetric component, which are uncorrelated; the skew-symmetric component is linear and depends only on the sum of the feature weights (Zielman & Heiser, 1996),. As already noted by Tversky (1977), the symmetric version of the Contrast Model is equivalent to a distinctive features model. Therefore, the symmetric component of the Contrast Model fits in the scheme of Figure 5.10, and has a network representation. The model recently proposed by Navarro and Lee (2004), like the Contrast Model, is a linear combination of common and distinctive features, with the specification that each feature enters either into a common features combination rule or into a distinctive features combination rule. Converting the common component into a distinctive component with the specifications in Equation 5.15 in this paper, we have an additive combination of two distinctive features models, which again is a distinctive features model in the total feature space. In fact, this hybrid type of model is an example of Carroll's (1976) general strategy of decomposing a (dis)similarity matrix into the sum of multiple trees or other graphical structures. Although the sum of two additive trees is not a tree, it

still is a distinctive features model (albeit perhaps not a parsimonious one). Finally, it is of interest to mention the possibility to combine these discrete structures with generalized context models and geometric prototype models (M. D. Lee & Navarro, 2002; Nosofsky & Zaki, 2002; Verbeemen, Storms, & Verguts, 2004; Zaki et al., 2003).

Several aspects of network representations allow statistical refinement. Given the feature structure, estimation of the feature weights is a rather standard statistical problem. Frank and Heiser (in press a) have shown how to determine standard errors and confidence intervals for the feature weights in the distinctive features model. When the data can be split up in a training and a testing sample, it is also possible to calculate statistical accuracy of parameter estimates, do model tests and find a well-balanced compromise between model fit and model complexity when the features are unknown (Frank & Heiser, in press b). Similar work has been done by M. D. Lee (2001) for additive clustering, Navarro and Lee (2001) for the Contrast Model, and Frank and Heiser (2005) for additive trees. The emergence of a full-fledged methodology for city-block models owes much to their additivity, the very same property that makes them such attractive models for psychological similarity and difference.

Chapter 6

Epilogue: General Conclusion and Discussion

6.1 Reviewing statistical inference in Feature Network Models

In this monograph, statistical inference in FNM has been accomplished using the multiple regression framework. It provided the basis for the estimation of standard errors, confidence intervals , model test and features subset selection. This framework has been helpful in solving some problems, but there remain problems unsolved. The following sections review the results.

Constrained estimation

Considering features in FNM as predictor variables leads to the univariate linear regression model with positivity constraints on the feature discriminability parameters. Due to these positivity constraints, the ordinary least squares estimator becomes the inequality constrained least squares estimator (ICLS). One of the key problems has been to assess the variability of the feature discriminability parameters estimated with the ICLS estimator, which has a truncated (normal) distribution. Statistical inference in inequality constrained least squares problems is far from straightforward. While there is abundant literature on the computational aspects of the ICLS estimators (*cf.* Golub & Loan, 1989; Lawson & Hanson, 1995; Wollan & Dykstra, 1987), a recent review on statistical inference in inequality constrained least squares problems (Sen & Silvapulle, 2002) showed that optimal estimators or tests of significance generally do not exist for such nonstandard methods. In this context, there is only one author (Liew, 1976) who proposed a direct method to obtain theoretical standard errors for the ICLS estimator. The combination of Liew's theory on standard errors for the ICLS estimator with an efficient algorithm to compute ICLS estimates (Algorithm AS 225, Wollan & Dykstra, 1987) written in Matlab code, made it possible to obtain feature discriminability values and their associated standard errors. This method can be used in any inequality constrained least squares situation.

Imposing positivity constraints results from a priori ideas about the true model or properties of the population. In the context of FNM the prior belief would be that

there exists a representation of the data in terms of a network where the edge lengths are all positive. If one believes that the true model is positive, then, negative values are non existing effects (in the sense that they only result from sampling error) and should lead to parameter values equal to zero. The theory about standard errors of inequality constrained least squares estimators proposed by Liew (1976) is based on the assumption that the true model parameters are positive and, subsequently, yields zero values for standard errors related to parameters where the constraints are activated. Negative effects, and therefore, non existing effects, yield standard errors equal to zero, which expresses the idea that non existing effects are not allowed to have variability. There are differences in opinion about this idea. Tibshirani (1996), for example, uses a ridge regression approximation for the estimation of standard errors for the Lasso parameters that also leads to zero values for the Lasso parameters that are shrunken to zero, and finds this an inconvenience. Shrinking parameters is a rather smooth procedure that eventually leads to parameter values equal to zero. Inequality constrained least squares is not smooth: a parameter is zero or positive. In the context of FNM, negative parameter values are assumed to result from sampling error (irreducible measurement error) and, consequently, should not be part of the model.

The theoretical standard errors for the feature discriminability parameters have been used in two different settings in this monograph. Chapter 2 provided results for the theoretical standard errors for an a priori known feature structure in FNM. Chapter 3 showed an application of the theoretical standard errors for the feature structure of additive trees, a special case of FNM. Standard errors that yielded adequate 95% t-confidence intervals were obtained in two situations: a priori known tree topologies and estimated tree topologies. The results on estimated tree topologies, which were based on the possibility to split the data in a training and a test sample, are expected to hold for the general FNM framework as well.

In both aforementioned studies, the performance of the theoretical standard errors was evaluated with Monte Carlo simulation techniques and compared with bootstrap standard deviations of the sampling distribution of the ICLS estimator. The performance criterion was the coverage probability of 95% t-confidence intervals. The coverage results for a priori known feature structure for the FNM were different for the theoretical standard errors compared to the bootstrap standard deviations, with a tendency to undercover for the bootstrap standard deviations and a tendency to overcover for the theoretical standard deviations. The tendency to undercover for the bootstrap confidence intervals was more prominent in the simulations with the additive tree models.

Although the bootstrap is renowned to be applicable in a wide range of situations, the inequality constrained least squares framework poses some limitations to the use of the bootstrap to assess the variance of a statistic. The consequence of imposing positivity constraints is that the empirical distribution is no longer centered around the true parameter value, which threatens the consistency of the bootstrap distribution and this might explain the tendency to undercover for the bootstrap confidence intervals . Especially when more constraints are activated, the distributions of the parameters are affected although it is not precisely known what the consequences are. Self and Liang (1987) studied several cases of constrained parameters

when true parameter values are on the boundary of parameter space. The authors approximated the distributions of the constrained parameters by a mixture of χ^2 distributions when a small number of parameters are constrained. With more activated constraints, as is often the case in additive trees, the distribution of the constrained parameters can no longer be approximated by a mixture of χ^2 distributions. The reason that the theoretical standard errors proposed by Liew (1976) work is that they are based on the standard errors of the unconstrained (ordinary) least squares estimator. According to Self and Liang (1987) in the presence of boundary parameters one could always reflect the parametric distributions across the boundary to create a larger problem where the boundary points become interior points. The ordinary least squares solution represents the larger problem in this case.

Bootstrap standard deviation

In this monograph, the theoretical standard errors were compared to the true values and to the bootstrap standard deviations. Unlike the computation of the theoretical standard deviations, which is straightforward once the correct formula is known, the calculation of the bootstrap standard deviations is preceded by several decisions and limitations. Considering the FNM as univariate regression models influenced the choice of a resampling method, but also the presence of dissimilarity values in FNM. In the context of regression models there are different ways of resampling with different outcomes (Efron & Tibshirani, 1998; Freedman, 1981; Freedman & Peters, 1984): sampling residuals or sampling pairs of observations (value of the dependent variable and corresponding row or the matrix of predictor variables). Bootstrapping pairs was imposed by the FNM setting as a way to avoid negative dissimilarities that could result from sampling residuals. It is also one of the resampling methods in the linear regression model context that is robust in presence of heterogeneous error variance (Liu & Singh, 1992b).

In addition to the choices that have to be made about the method of resampling, the properties of the bootstrap standard deviations are not completely understood yet. Even in the "simple" linear regression case, without constraints, the consistency of the variance of the bootstrap distribution has not received much attention, while the consistency of the OLS estimator is well established in the literature; Gonçalves and White (2005) say on this topic:

> "The consistency of the bootstrap distribution, however, does not guarantee the consistency of the variance of the bootstrap distribution (the bootstrap variance) as an estimator of the asymptotic variance, because it is well known that convergence in distribution of a random sequence does not imply convergence of moments".

A better evaluation of the bootstrap standard deviations themselves could be achieved by performing a double bootstrap, as suggested by Gonçalves and White (2005) where the bootstrap is used to simulate the distribution of the t statistic which is based on a standard deviation that in turn has been estimated by the bootstrap. However, the authors did not actually perform the double bootstrap, which means

bootstrapping the bootstrap, because the implementation is extremely computationally intensive. There is room for further improvement of bootstrap standard deviations in the case of inequality constraints.

Assumptions and limitations

The application of the theoretical standard errors and associated 95% t-confidence intervals is limited to the assumption of normally distributed error terms. It is well known that the assumption of normality does not always hold in psychological research (Micceri, 1989). In the context of FNM, each dissimilarity value typically represents the mean of the dissimilarity values obtained from N subjects, and by the central limit theorem it is to be expected that the mean dissimilarity values become approximately normally distributed fairly rapidly. A more challenging problem is the issue of correlated data: "the future of linear models research lies primarily in developing methods for correlated data" (Christensen, 2002). The problem of dependency in the data has not been addressed in this monograph, but is likely to occur in the FNM because error terms associate with dissimilarity values that share the same objects are possibly correlated. The data used in the simulation studies were generated under the assumptions of normality, independence and homogeneity.

For the special case of FNM, the specification of the error structure might be a difficult task to accomplish in practice. There are specific experimental settings that inevitably produce data that yield correlated residuals, also called unobserved heterogeneity, as in longitudinal or multilevel data (*cf.* Skrondal & Rabe-Hesket, 2004). When individuals are clustered, for example students in classes, or when the same individuals are measured several times in a longitudinal setting, the residuals become correlated and the error structure is reasonably predictable. In the context of FNM, or more generally, dissimilarity matrices, the error structure is not precisely known. In addition, it is not clear how to assess the amount of dependency present in the data and the available tests are limited to specific settings, not comparable to the situation in FNM. The Durbin-Watson test (Durbin & Watson, 1950) is intended for autoregressive residuals and other tests like the Box test (Box, 1949) and the intraclass correlation (*cf.* Stevens, 1992) are useful to test the independence assumptions in the presence of several groups of individuals.

Given the difficulty to specify the error structure and the lack of adequate tests, the question comes up whether it is useful to adjust for correlated residuals a posteriori. Unlike the longitudinal studies, where dependency is inevitable, the FNM setting offers possibilities to reduce the occurrence of correlated error terms. For example, taking the mean of the dissimilarity values from a substantial number of subjects already reduces the correlation. Or one could use permutation techniques on the collected data matrices to disentangle the correlation structure between dissimilarity values that share the same object. More research is necessary to specify the error structure and to find adequate methods to prevent dependency in dissimilarity data.

If it is not possible to prevent dependency during the data collection step, one could use generalized least squares to take into account error correlation, but it would not solve the constrained estimation problem. The challenge is to combine in-

equality constrained least squares with generalized least squares. Very few attempts have been made to obtain estimates for the generalized inequality constrained least squares estimator, GICLS (Werner, 1990; Werner & Yapar, 1996). Assessing the variability of the GICLS estimator is still a far way to go, although Gulliksson and Wedin (2000) obtained some results in the perturbation theory for GICLS that are useful to assess the stability of the solution.

It should be noted that violations of the independence assumption not only affect parametric statistical inference (theoretical standard errors), but also affect the bootstrap. This means that the bootstrap method needs to be adjusted. Künsch (1989) and Liu and Singh (1992a) introduced the moving blocks bootstrap for use with dependent data. Gonçalves and White (2005) have further refined this method for the linear models with dependent data by establishing conditions for the consistency of the moving blocks bootstrap estimators of the variance of the least squares estimator. Using this bootstrap method might improve the assessment of bootstrap standard deviations for the feature discriminability parameters in FNM, although the results might not be the same for the ICLS estimator, and needs further research.

Another issue that has not been addressed in this monograph, is the problem of multiple confidence intervals. If confidence intervals are used to decide which features are important (especially the additive tree models have a large number of features), it eventually leads to the problem of multiple testing. A way out could be to use the Positive Lasso to select the best subset of features. In several applications in this monograph, the feature set selected by the Positive Lasso corresponded to the set of features with appropriate confidence intervals. There obviously exists a link between the two methods, although it has not been demonstrated yet. Furthermore, there are promising results available for the extension of the LARS algorithm to generalized linear models (see the discussion of Efron et al., 2004), which might be a solution to possible correlated error terms in the FNM. For use with the additive trees models, the Positive Lasso needs further adjustments because the feature structure is more restricted than in the general FNM. The Positive Lasso and the Lasso both have the additional advantage of being robust to correlated predictors, or multicollinearity.

6.2 Features and graphical representation

The set of distinctive features

The representation of features with Gray codes proved to be useful in several aspects. In a practical sense, the representation of features by the Gray code considerably simplifies computer manipulations of feature sets. The convenient attribute of Gray codes to represent features by a rank number (a simple integer) saves computer time and memory because the original feature set can be retrieved by simply keeping track of the corresponding rank number. Another advantage of the representation of features by a unique rank number is the possibility to get back the original features after transformations to featurewise distances, which is the transformation from the objects $\times$ features matrix $\mathbf{E}$ to the pairs of objects $\times$ features matrix $\mathbf{X}$. This transformation is not reversible because the results are not unique. The practical properties

of the Gray code rank numbers are particularly useful in the generation of the feature sets, for efficient storage during Monte Carlo simulations, but also for efficient comparison of different tree topologies. In a more conceptual sense, the representation in Gray codes allows for defining a finite solution space, which can be further reduced to distinctive features only, through the transformation from **E** to **X**. This transformation limits the search for predictors to the set of truly distinctive features and increases the gain of using Gray coding.

Given that it is possible to generate the complete set of distinctive features for up to 22 objects in an efficient way, it is tempting to search for the optimal set of features for a given data set. Instead, the Positive Lasso selects a suboptimal solution that has better generalizability properties. This combination of the Positive Lasso algorithm and the complete set of distinctive features, has several additional advantages over the existing algorithms. It selects the best subset regardless of the number of features and avoids deciding on the number of features a priori. The selected subset is not biased toward a certain graphical representation because all possible feature structures are allowed. However, the method needs to be improved for data sets that have more than 22 objects or stimuli.

FNM is restricted to the use of distinctive features, which has some computational advantages as discussed in the previous paragraphs. In the introduction to this monograph, the distinctive features were presented as opposed to common features. The Contrast Model (Tversky, 1977) combines both types of features, but most of the models based on features that were developed later, mainly concentrate on one type of feature leading to common features models (CF) or distinctive features models (DF). The possibility to transform the CF model into the DF model and vice versa, has already been demonstrated by Sattath and Tversky (1987) and Carroll and Corter (1995). Chapter 5 further refined the transformation from CF to DF by showing that for any fitted CF model it is possible to find an equally well fitting DF model with the same set of shared features (common features) and associated feature weights, while keeping the same number of parameters. In this transformation, the CF model is a special case of the distinctive features model, which is a new result. Within this framework, a model that combines common and distinctive features can be represented as a sum of two separate DF models. However, the opposite transformation, from DF to CF is only possible if the objects are equidistant from the origin.

Network representation of features

Compared to all the models that are based on feature structuress, the graphical representation in terms of a network is unique for FNM. The network representation of features offers an interesting framework for theoretical comparison and practical use of several scaling and clustering methods. Figure 5.10 in Chapter 5 has shown that a whole family of discrete models of similarity are in fact special cases of the distinctive features model. The distinctive features models themselves are special cases of the city-block model and result from the restriction that the coordinate values be binary. Chapter 5 demonstrated that a coordinate system is not always necessary to represent city-block models. The additivity properties of the city-block distances

allow for dropping the whole coordinate system and as a consequence, the dissimilarities can equally well be embedded in a network.

Embedding the network

To represent FNM, which are in general high-dimensional structures, as low dimensional feature graphs it is necessary to considerably reduce the dimensionality of the space. In this monograph, all network representations were obtained with multidimensional scaling performed with PROXSCAL[1]. The input distances were Euclidean distances computed with the feature discriminability parameters, and they were usually represented in 2-dimensional space. To represent the features in the same solutions space, PROXSCAL offers the possibility to constrain the solution space by a linear combination of the feature variables. As a result, the features can be represented as vectors leading from the origin through the point with coordinates equal to the correlations of each feature with each dimension. For ease of interpretation, the centroids of the objects that possess a particular feature can be projected onto the vector representing that feature. The same can be done for the objects that are not characterized by that feature. Labeling these projected points with plus and minus signs gives insight in the feature patterns of the objects.

The embedding of feature graphs in a lower dimensional space is for display purposes only because the model is specified by the network structure, the feature discriminability parameters and the model fit (the goodness-of-fit between the data and the reconstructed network distance). The fit between the network distance and the distance in the visual display, the embedding fit, is of secondary importance. The embedding is somewhat arbitrary because there are many possibilities to achieve this goal, depending on the distance transformations used or the type of start configuration used. In addition, the embedding is not restricted to the use of multidimensional scaling on the feature distances. Without using the feature distances, the objects and the features could also be represented in a biplot obtained with correspondance analysis.

Figure 6.1 shows an example of a plot representing the 14 features that characterize the presidents of the United States (the data were described in the introductory chapter), obtained with correspondance analysis, using row principal normalization. In row-principal normalization, a president point is located in the center of gravity of the features that he possesses. By connecting each president point with his own feature points, it is possible to reconstruct the feature graph from the correspondence analysis plot. However, in contrast to the feature network representation of the same data (See Figure 1.1, Chapter 1), the correspondence analysis plot does not allow a direct reconstruction of the distances between presidents. The strong point of the feature network representation results from the possibility to represent the feature structure as well as the distances between the objects by labeling the edges with the corresponding feature distances.

[1] PROXSCAL is a multidimensional scaling program distributed as part of the Categories package by SPSS, Meulman & Heiser, 1999

FNM and tree representations

Chapter 3 showed that given a special, nested, feature structure, formed by a combination of cluster features, unique features and internal nodes, the feature network representation becomes an additive tree representation. Do the results obtained for the additive trees also apply to hierarchical trees? Not directly, because in the hierarchical tree additional constraints are necessary to obtain equal distances to the root. In our context this means that extra constraints should be imposed on the feature discriminability parameters for the unique features. The problem could however be circumvented by using common features. The hierarchical tree can be obtained from a common feature model without unique features, which means that no extra constraints are necessary.

In the psychological literature, the relation between features and tree representations exists for a long time, while this relation is unknown in the phylogenetic tree domain. Both research areas might benefit from their mutual results. In particular, the use of features in FNM along with the univariate multiple regression framework led to two results that might be of interest for phylogenies. The first one is the possibility to use the *generalized cross-validation* statistic as an estimate of prediction error. This convenient closed form formula can be used to compare different tree topologies, even if both topologies have the same number of degrees of freedom. Being able to compare tree topologies with the same number of degrees of freedom, is an advantage over the likelihood ratio test that is commonly used to compare tree topologies but is limited to the case of nested topologies. The second result concerns the possibility of using cluster features to test in an easy way, events of speciation, the evolutionary process by which new biological species arise. Further improvements of statistical inference in the additive tree representations of FNM could be obtained by using the Positive Lasso to prune the tree, instead of using confidence intervals to select the relevant set of features. Pruning the tree will necessitate modifications of the present implementation of the Positive Lasso to simplify the tree structure in specific areas in order to keep the representation tree-shaped.

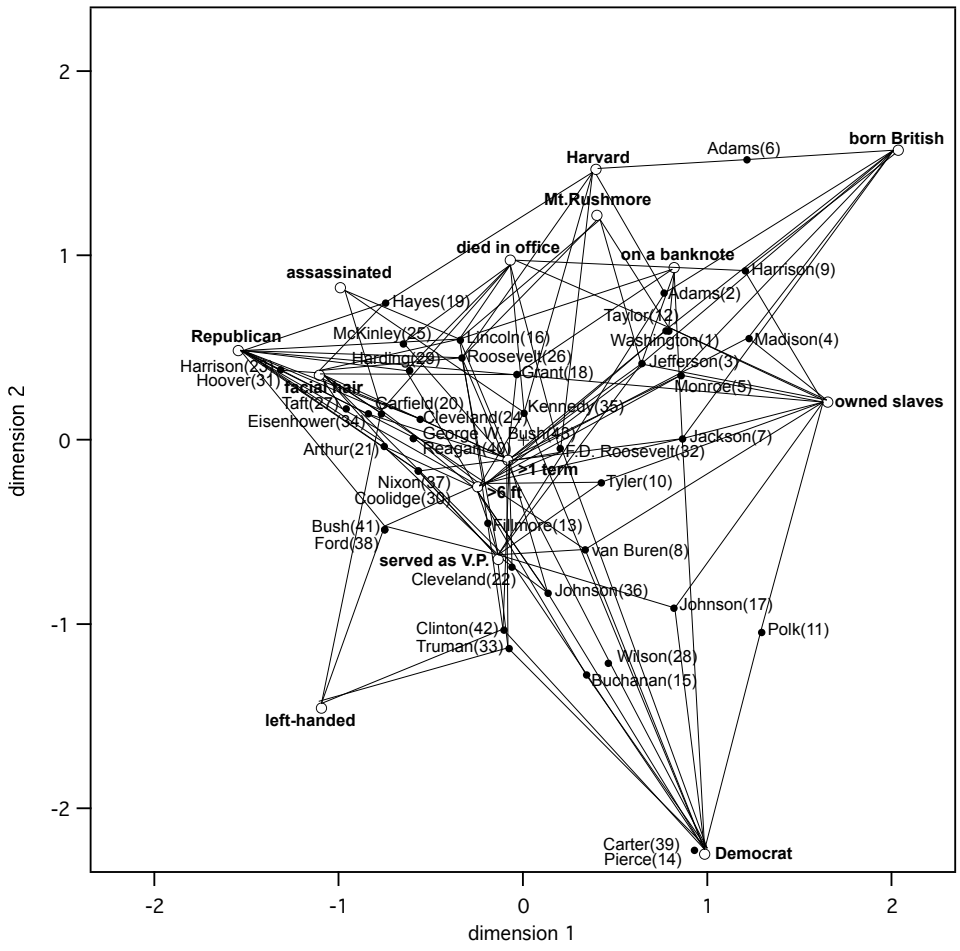

Figure 6.1: Biplot in 2 dimensions obtained with correspondence analysis of the 14 features describing the 43 presidents of the United States. The presidents are linked with the features they possess. (Normalization: row principal).

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716- 723.

Arabie, P. (1991). Was Euclid an unnecessarily sophisticated psychologist? *Psychometrika*, *56*, 567-587.

Arabie, P., & Carroll, J. D. (1980). MAPCLUS: A mathematical programming approach to fitting the ADCLUS model. *Psychometrika*, *45*, 211-235.

Ashby, F., & Maddox, W. (1990). Integrating information from separable dimensions. *Journal of Experimental Psychology: Human Perception & Performance*, *16*, 598-612.

Attneave, F. (1950). Dimensions of similarity. *American Journal of Psychology*, *63*, 516-556.

Backhaus, W., Menzel, R., & Kreißl, S. (1987). Multidimensional scaling of color similarities in bees. *Biological Cybernetics*, *56*, 293-304.

Barthélemy, J. P., & Guénoche, A. (1991). *Trees and Proximity Representations*. New York: Wiley.

Björk, A. (1996). *Numerical methods for least squares problems*. Philadelphia, PA.: SIAM.

Borg, I., & Leutner, D. (1983). Dimensional models for the perception of rectangles. *Perception & Psychophysics*, *34*, 257-267.

Bortz, J. (1974). Kritische Bemerkungen ueber den Einsatz nicht-euklidischer Metriken im Rahmen der multidimensionalen Skalierung [Critical remarks on the use of non-Euclidean metrics in the context of multidimensional scaling]. *Archiv für Psychologie*, *126*, 196-212.

Box, G. E. P. (1949). A general distribution theory for a class of likelihood criteria. *Biometrika*, *36*, 317-346.

Brusco, M. (2001). A simulated annealing heuristic for unidimensional and multidimensional (City-Block) scaling of symmetric proximity matrices. *Journal of Classification*, *18*, 3-33.

Brusco, M. (2002). Integer programming methods for seriation and unidimensional scaling of proximity matrices: A review and some extensions. *Journal of Classification*, *19*, 45-67.

Buja, A., & Swayne, D. (2002). Vizualization methodology for multidimensional scaling. *Journal of Classification*, *19*, 7-43.

Bulmer, M. (1991). Use of the method of generalized least squares in reconstructing phylogenies from sequence data. *Molecular Biology and Evolution*, *8*, 868-883.

Buneman, P. (1971). The recovery of trees from measures of dissimilarity. In F. R.

Hodson, D. G. Kendall, & P. Tautu (Eds.), *Mathematics in the archeological and historical sciences* (p. 387-395). Edinburgh: University Press.

Buneman, P. (1974). A note on the metric properties of trees. *Journal of Combinatorial Theory*, *17*, 48-50.

Busemann, H. (1955). *The geometry of geodesics*. New York: Academic Press (unabridged republication in 2005 by Dover Press, Mineola, New York).

Carroll, J. D. (1976). Spatial, non-spatial and hybrid models for scaling. *Psychometrika*, *41*(4), 439-463.

Carroll, J. D., & Arabie, P. (1983). INDCLUS: An individual differences generalization of the ADCLUS model and the MAPCLUS algorithm. *Psychometrika*, *48*, 157-169.

Carroll, J. D., & Chang, J. J. (1973). A method for fitting a class of hierarchical tree structure models to dissimilarities data and its application to some "body parts" data of Miller's. In *Proceedings of the 81st annual convention of the american psychological association* (Vol. 8, p. 1097-1098).

Carroll, J. D., Clark, L. A., & DeSarbo, W. S. (1984). The representation of three-way proximity data by single and multiple tree structure models. *Journal of Classification*, *1*, 25-74.

Carroll, J. D., & Corter, J. E. (1995). A graph-theoretic method for organizing overlapping clusters into trees, multiple trees, or extended trees. *Journal of Classification*, *12*, 283-313.

Carroll, J. D., & Pruzansky, S. (1980). Discrete and hybrid scaling models. In E. D. Lanterman & H. Feger (Eds.), *Similarity and Choice* (p. 108-139). Bern: Hans Huber.

Chasles, M. (1875). *Aperçu historique sur l'origine et le développement des méthodes en géométrie : particulièrement de celles qui se rapportent a la géométrie moderne suivi d'un mémoire de géométrie sur deux principes généraux de la science, la dualité et l'homographie*. Paris: Gauthier-Villars.

Chaturvedi, A., & Carroll, J. D. (1994). An alternating combinatiorial optimization approach to fitting the INDCLUS and generalized INDCLUS models. *Journal of Classification*, *11*, 155-170.

Christensen, R. (2002). Linear and log-linear models. In A. E. Raftery, M. A. Tanner, & M. T. Wells (Eds.), *Statistics in the 21st Century* (p. 319-325). Boca Raton: Chapman & Hall.

Colonius, H., & Schulze, H. (1981). Tree structures for proximity data. *British Journal of Mathematical and Statistical Psychology*, *34*, 167-180.

Corter, J. E. (1982). ADDTREE/P: A PASCAL program for fitting additive trees based on Sattath & Tversky's ADDTREE algorithm. *Behavior Research Methods and Instrumentation*, *14*, 353-354.

Corter, J. E. (1996). *Tree models of similarity and association*. Thousand Oaks: SAGE Publications.

Corter, J. E. (1998). An efficient metric combinatorial algorithm for fitting additive trees. *Multivariate Behavioral Research*, *33*, 249-272.

Corter, J. E., & Tversky, A. (1986). Extended similarity trees. *Psychometrika*, *51*, 429-451.

Cross, D. (1965). Metric properties of multidimensional stimulus generalization.

In D. Mostofsky (Ed.), *Stimulus generalization* (p. 72-93). Stanford: Stanford University Press.

Cunningham, J. P. (1974, August). *Finding an optimal tree realization of a proximity matrix.* Paper presented at the Mathematical Psychology Meeting, Ann Arbor.

Cunningham, J. P. (1978). Free trees and bidirectional trees as representations of psychological distance 19. *Journal of Mathematical Psychology*, *17*, 165-188.

Davey, B., & Priestley, H. (2002). *Introduction to Lattices and Order*. Cambridge, UK: Cambridge University Press.

De Soete, G. (1983). A least squares algorithm for fitting additive trees to proximity data. *Psychometrika*, *48*, 621-626.

De Soete, G., & Carroll, J. D. (1996). Tree and other network models for representing proximity data. In P. Arabie, L. J. Hubert, & G. De Soete (Eds.), *Clustering and classification.* River Edge, NJ: World Scientific Publishing.

Draper, N. R., & Smith, H. (1998). *Applied Regression Analysis*. New York: Wiley.

Durbin, J., & Watson, G. S. (1950). Testing for serial correlation in least squares regression: I. *Biometrika*, *37*, 409-428.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. J. (2004). Least Angle Regression. *The Annals of Statistics*, *32*, 407-499.

Efron, B., & Tibshirani, R. J. (1998). *An introduction to the Bootstrap*. Boca Raton: CRC Press LLC.

Eisler, H. (1973). The algebraic and statistical tractability of the city-block metric. *British Journal of Mathematical and Statistical Psychology*, *26*, 212-218.

Eisler, H., & Roskam, E. (1977). Multidimensional similarity: An experimental and theoretical comparison of vector, distance, and set-theoretical models. *Acta Psychologica*, *41*, 1-46 and 335-363.

Emde, G. Von der, & Ronacher, B. (1994). Perception of electric properties of objects in electrolocating weakly electric fish: Two-dimensional similarity scaling reveals a City-Block metric. *Journal of Comparative Physiology A*, *175*, 801-812.

Felsenstein, J. (1983). Statistical inference of phylogenies. *Journal of the Royal Statistical Society: Series A (General)*, *146*, 246-272.

Felsenstein, J. (1985). Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, *39*, 783-791.

Felsenstein, J. (2004). *Inferring Phylogenies*. Sunderland, Massachusetts: Sinauer Associates.

Flament, C. (1963). *Applications of graph theory to group structure*. Englewood Cliffs, New Jersey: Prentice-Hall.

Frank, L. E., & Heiser, W. J. (2004, June). *Statistical inference in Feature Network Models and additive trees.* Paper presented at the International Meeting of the Psychometric Society, Pacific Grove, CA.

Frank, L. E., & Heiser, W. J. (2005). Standard errors, prediction error and model tests in additive trees. Manuscript submitted for publication.

Frank, L. E., & Heiser, W. J. (in press a). Estimating standard errors in Feature Network Models. *British Journal of Mathematical and Statistical Psychology*.

Frank, L. E., & Heiser, W. J. (in press b). Feature selection in Feature Network Models: finding predictive subsets of features with the Positive Lasso. *British Journal of Mathematical and Statistical Psychology*.

Freedman, D. A. (1981). Bootstrapping regression models. *The Annals of Statistics*, *9*, 1218-1228.

Freedman, D. A., & Peters, S. C. (1984). Bootstrapping a regression equation: some empirical results. *Journal of the American Statistical Association*, *79*, 97-106.

Friedman, J. H., & Popescu, B. E. (2006). Gradient directed regularization for linear regression and classification. Submitted manuscript.

Galanter, E. H. (1956). An axiomatic and experimental study of sensory order and measure. *Psychological Review*, *63*(1), 16-28.

Gardner, M. (1972). The curious properties of the Gray code and how it can be use to solve puzzles. *Scientific American*, *227*, 106-109.

Garner, W. (1974). *The processing of information and structure*. New York: Wiley.

Gascuel, O. (1994). A note on Sattath and Tversky's, Saitou and Nei's, and Studier and Keppler's algorithms for inferring phylogenies from evolutionary distances. *Molecular Biology and Evolution*, *11*, 961-963.

Gascuel, O., & Levy, D. (1996). A reduction algorithm for approximating a (non-metric) dissimilarity by a tree distance. *Journal of Classification*, *13*, 129-155.

Gati, I., & Tversky, A. (1982). Representations of qualitative and quantitative dimensions. *Journal of Experimental Psychology: Human Perception and Performance*, *8*, 325-340.

Gati, I., & Tversky, A. (1984). Weighting common and distinctive features in perceptual and conceptual judgments. *Cognitive Psychology*, *16*, 341-370.

Gilbert, E. N. (1958). Gray codes and paths on the *n*-cube. *Bell Systems Technical Journal*, *37*, 815-826.

Goldstone, R. (1994). Influences of categorization on perceptual discrimination. *Journal of Experimental Psychology: General*, *123*, 178-200.

Golub, G. H., & Loan, C. F. van. (1989). *Matrix Computations* (second ed.). Baltimore: The Johns Hopkins University Press.

Gonçalves, S., & White, H. (2005). Bootstrap standard error estimates for linear regression. *Journal of the American Statistical Association*, *100*, 970-979.

Goodman, N. (1951). *The Structure of Appearence*. Indianapolis, Indiana: Bobbs-Merrill.

Goodman, N. (1977). *The Structure of Appearence* (3rd ed.). Dordrecht, Holland: Reidel.

Gray, F. (1953, March). Pulse code communications. U.S. Patent 2632058.

Groenen, P. J. F., & Heiser, W. J. (1996). The tunneling method for global optimization in multidimensional scaling. *Psychometrika*, *61*(3), 529-550.

Groenen, P. J. F., Heiser, W. J., & Meulman, J. J. (1998). City-block scaling: Smoothing strategies for avoiding local minima. In Balderjahn, R. R. Mathar, & M. Schader (Eds.), *Classification, data analysis and data highways* (p. 46-53). Berlin: Springer.

Groenen, P. J. F., Heiser, W. J., & Meulman, J. J. (1999). Global optimization in least-squares multidimensional scaling by distance smoothing. *Journal of Classification*, *16*, 225-254.

Gulliksson, M., & Wedin, P.-Å. (2000). Perturbation theory for generalized and constrained linear least squares. *Numerical Linear Algebra with Applications*, *7*, 181-195.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157-1182.

Hastie, T., Tibshirani, R. J., & Friedman, J. H. (2001). *The Elements of Statistical Learning; Data mining, Inference, and Prediction*. New York: Springer Verlag.

Hays, W. (1958). An approach to the study of trait implication and trait similarity. In R. Tagiuri & L. Petrullo (Eds.), *Person perception and interpersonal behavior* (p. 289-299). Stanford: Stanford University Press.

Heiser, W. J. (1989). The city-block model for three-way multidimensional scaling. In R. Coppi & S. Bolasco (Eds.), *Multiway data analysis* (p. 395-404). Amsterdam: North-Holland.

Heiser, W. J. (1991). A generalized majorization method for least squares multidimensional scaling of pseudodistances that may be negative. *Psychometrika*, *56*(1), 7-27.

Heiser, W. J. (1998). Fitting graphs and trees with multidimensional scaling methods. In C. Hayashi, N. Ohsumi, K. Yajima, Y. Tanaka, H.-H. Bock, & Y. Baba (Eds.), *Data science, classification and related methods* (p. 52-62). Tokyo: Springer Verlag.

Heiser, W. J., & Busing, F. M. T. A. (2004). Multidimensional scaling and unfolding of symmetric and asymmetric proximity relations. In D. Kaplan (Ed.), *The SAGE handbook of quantitative methodology for the social sciences* (p. 25-48). Thousand Oaks, CA: Sage.

Heiser, W. J., & Frank, L. E. (2005). On the relationship between the distinctive features model and the common features model. unpublished manuscript.

Heiser, W. J., & Meulman, J. J. (1997). Representation of binary multivariate data by graph models using the hamming metric. In E. Wegman & S. Azen (Eds.), *Computing science and statistics* (Vol. 29 (2), p. 517-525). Fairfax, VA: Interface Foundation of North America.

Hubert, L. J., & Arabie, P. (1988). Relying on necessary conditions for optimization: Unidimensional scaling and some extensions. In H. H. Bock (Ed.), *Classification and related methods of data analysis* (p. 463-472). Amsterdam: North-Holland.

Hubert, L. J., Arabie, P., & Hesson-Mcinnis, M. (1992). Multidimensional scaling in the city-block metric: a combinatorial approach. *Journal of Classification*, *9*, 211-136.

Hubert, L. J., Arabie, P., & Meulman, J. J. (2001). *Combinatorial Data Analysis: Optimization by Dynamic Programming*. Philadelphia, NJ: SIAM.

Huelsenbeck, J. P., & Rannala, B. (1997). Phylogenetic methods come of age: testing hypotheses in an evolutionary context. *Science*, *276*, 227-232.

Jain, A., & Dubes, R. (1988). *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice Hall.

Jakobson, R., Fant, C. G. M., & Halle, M. (1965). *Preliminaries to speech analysis. The distinctive features and their correlates. (Reprint of the original edition of 1953).* Cambridge, MA: MIT Press.

Joly, S., & Le Calvé, G. (1994). Similarity functions. In B. van Cutsem (Ed.), *Classification and Dissimilarity Analysis* (p. 67-86). New York: Springer Verlag.

Kamada, T., & Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters*, *31*, 7-15.

Keren, G., & Baggen, S. (1981). Recognition models for alphanumeric characters. *Perception & Psychophysics*, *29*, 234-246.

Kishino, H., & Hasegawa, M. (1990). Converting distance to time: application to human evolution. *Methods in Enzymology*, *183*, 550-570.

Klatzky, R., Beall, A., Loomis, J., Golledge, R., & Philbeck, J. (1999). Human navigation ability: Tests of the encoding error model of path integration. *Spatial Cognition and Computation*, *1*, 31-65.

Klauer, K. C. (1989). Ordinal network representation: representing proximities by graphs. *Psychometrika*, *54*(4), 737-750.

Klauer, K. C. (1994). Representing proximities by network models. In E. Diday (Ed.), *New approaches in classification and data analysis.* Heidelberg: Springer Verlag.

Klauer, K. C., & Carroll, J. D. (1989). A mathematical programming approach to fitting general graphs. *Journal of Classification*, *6*, 247-270.

Krackhardt, D. (1988). Predicting with networks: nonparametric multiple regression analysis of dyadic data. *Social Networks*, *10*, 359-381.

Kruschke, J. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review*, *99*, 22-44.

Kuennapas, T., & Janson, A.-J. (1969). Multidimensional similarity of letters. *Perceptual and Motor Skills*, *28*, 3-12.

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, *17*, 1217-1241.

Landahl, H. (1945). Neural mechanisms for the concepts of difference and similarity. *Bulletin of Mathematical Biophysics*, *7*, 83-88.

Lawson, C. L., & Hanson, R. J. (1995). *Solving least squares problems*. Philadelphia: Society for Industrial and Applied Mathematics.

Le Calvé, G. (1985). Distances à centre [center distances]. *Statistiques et Analyse des Données*, *10*, 29-44.

Lee, M. D. (2001). On the complexity of additive clustering models. *Journal of Mathematical Psychology*, *45*, 131-148.

Lee, M. D., & Navarro, D. J. (2002). Extending the ALCOVE model of category learning to featural stimulus domains. *Psychonomic Bulletin & Review*, *9*, 43-58.

Lee, W. C., & Rodgers, J. L. (1998). Bootstrapping correlation coefficients using univariate and bivariate sampling. *Psychological Methods*, *3*, 91-103.

Li, W.-H. (1989). A statistical test of phylogenies estimated from sequence data. *Molecular Biology and Evolution*, *6*, 424-435.

Liew, C. K. (1976). Inequality constrained least-squares estimation. *Journal of the American Statistical Association*, *71*, 746-751.

Liu, R. Y., & Singh, K. (1992a). Moving blocks jackknife and bootstrap capture weak dependence. In R. LePage & L. Billiard (Eds.), *Exploring the Limits of the Bootstrap* (p. 225-248). New York: Wiley.

Liu, R. Y., & Singh, K. (1992b). Efficiency and robustness in resampling. *The Annals of Statistics*, *20*, 370-384.

MacKay, D. (2001). Probabilistic multidimensional scaling using a city-block metric. *Journal of Mathematical Psychology*, *45*(3), 249-264.

Mallows, C. L. (1973). Some comments on C_p. *Technometrics*, *15*, 661-675.

Mallows, C. L. (1995). More comments on C_p. *Technometrics*, *37*, 362-372.

Martinez, W. L., & Martinez, A. R. (2002). *Computational statistics handbook with Matlab*. Boca Raton: Chapman and Hall.

Melara, R., Marks, L., & Lesko, K. (1992). Optional processes in similarity judgments. *Perception & Psychophysics*, *51*, 123-133.

Meulman, J. J., & Heiser, W. J. (1999). *Categories*. Chicago: SPSS.

Micceri, T. (1989). The unicorn, the normal curve, and other improbable creatures. *Psychological Bulletin*, *105*, 156-166.

Micko, H. C., & Fischer, W. (1970). The metric of multidimensional psychological spaces as a function of differential attention to subjective attributes. *Journal of Mathematical Psychology*, *7*, 118-143.

Miller, G. A. (1969). Psychological method to investigate verbal concepts. *Journal of Mathematical Psychology*, *6*, 169-191.

Miller, G. A., & Nicely, P. E. (1955). An analysis of perceptual confusions among some english consonants. *Journal of the Acoustical Society of America*, *27*, 338-352.

Mirkin, B. (1987). Additivie clustering and qualitative factor analysis methods for similarity matrices. *Journal of Classification*, *6*, 247-270.

Mirkin, B. (1990). A sequential fitting procedure for linear data analysis models. *Journal of Classification*, *7*, 167-195.

Mirkin, B. (1996). Clustering for contingency tables: boxes and partitions. *Statistics and Computing*, *6*, 217-229.

Mirkin, B. (1998). Least squares structuring, clustering, and data processing issues. *The Computer Journal*, *41*, 518-536.

Mooijaart, A., van der Heijden, P. G. M., & van der Ark, L. (1999). A least squares algorithm for a mixture model for compositional data. *Computational Statistics & Data Analysis*, *30*, 359-379.

Navarro, D. J., & Lee, M. D. (2001). Clustering using the contrast model. In J. Moore & K. Stenning (Eds.), *Proceedings of the 23rd annual conference of the cognitive science society* (p. 686-691). Mahwah, NJ: Lawrence Erlbaum.

Navarro, D. J., & Lee, M. D. (2004). Common and distinctive features in stimulus similarity: A modified version of the contrast model. *Psychonomic Bulletin & Review*, *11*, 961-974.

Nei, M., Stephens, J. C., & Saitou, N. (1985). Methods for computing the standard errors of branching points in an evolutionary tree and their application to molecular data from human and apes. *Molecular Biology and Evolution*, *2*, 66-85.

Nijenhuis, A., & Wilf, H. S. (1978). *Combinatorial algorithms for computers and calculators*. New York: Academic Press.

Nosofsky, R. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *10*, 104-114.

Nosofsky, R. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, *115*, 39-57.

Nosofsky, R. (1987). Attention and learning processes in the identification and categorization of integral stimuli. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *13*, 87-108.

Nosofsky, R. (1992). Exemplars, prototypes, and similarity rules. In A. Healy, S. Kosslyn, & R. Shiffrin (Eds.), *From learning theory to connectionist theory: Es-*

says in honour of william estes, vol. 1 (p. 149-167). Hillsdale, NJ: Lawrence Erlbaum.

Nosofsky, R., & Zaki, S. (2002). Exemplar and prototype models revisited: Response strategies, selective attention, and stimulus generalization. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *28*, 924-940.

Okada, A., & Imaizumi, T. (1980). Nonmetric method for extended INDSCAL model. *Behaviormetrika*, *7*, 13-22.

Ota, R., Waddell, P. J., Hasegawa, M., Shimodaira, H., & Kishino, H. (2000). Appropriate likelihood ratio tests and marginal distributions for evolutionary tree models with constraints on parameters. *Molecular Biology and Evolution*, *17*, 798-803.

Parault, S., & Schwanenflugel, P. (2000). The development of conceptual categories of attention during the elementary school years. *Journal of Experimental Child Psycholog*, *75*, 245-262.

Pruzansky, S., Tversky, A., & Carroll, J. D. (1982). Spatial versus tree representations of proximity data. *Psychometrika*, *47*(1), 3-19.

Ramsay, J. O. (1982). Some statistical approaches to multidimensional scaling data. *Journal of the Royal Statistical Society. Series A (General)*, *145*(3), 285-312.

Restle, F. (1959). A metric and an ordering on sets. *Psychometrika*, *24*, 207-220.

Restle, F. (1961). *Psychology of judgment and choice*. New York: Wiley.

Ronacher, B. (1992). Pattern recognition in honeybees: Multidimensional scaling reveals city-block metric. *Vision Research*, *32*, 1837-1843.

Rosenberg, S., & Kim, M. P. (1975). The method of sorting as a data-gathering procedure in multivariate research. *Multivariate Behavioral Research*, *10*(4), 489-502.

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired-associate learning tasks. *Journal of Experimental Psychology*, *53*, 94-101.

Rzhetsky, A., & Nei, M. (1992). A simple method for estimating and testing minimum-evolution trees. *Molecular Biology and Evolution*, *9*, 945-967.

Saitou, N., & Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, *4*, 406-425.

Sattath, S., & Tversky, A. (1977). Additive similarity trees. *Psychometrika*, *42*, 319-345.

Sattath, S., & Tversky, A. (1987). On the relation between common and distinctive feature models. *Psychological Review*, *94*(1), 16-22.

Savage, C. (1997). A survey of combinatorial Gray codes. *SIAM Review*, *39*, 605-629.

Schott, B. (2003). *Schott's Original Miscellany*. New York: Bloomsbury.

Self, S. G., & Liang, K. Y. (1987). Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association*, *82*, 605-610.

Sen, P. K., & Silvapulle, M. J. (2002). An appraisal of some aspects of statistical inference under inequality constraints. *Journal of Statistical Planning and Inference*, *107*, 3-43.

Shepard, R. N. (1964). Attention and the metric structure of the stimulus space. *Journal of Mathematical Psychology*, *1*, 54-87.

Shepard, R. N. (1972). Psychological representation of speech sounds. In E. E.

David Jr. & P. Denes (Eds.), *Human communication: A unified view* (p. 67-113). New York: McGraw-Hill.

Shepard, R. N. (1974). Representation of structure in similarity data: Problems and prospects. *Psychometrika*, *39*, 373-421.

Shepard, R. N. (1980). Multidimensional scaling, tree-fitting, and clustering. *Science*, *210*, 390-398.

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, *237*, 1317-1323.

Shepard, R. N. (1991). Integrality versus separability of stimulus dimensions: From an early convergence of evidence to a proposed theoretical basis. In G. Lockhead & J. Pomerantz (Eds.), *The perception of structure* (p. 53-71). Washington, DC: APA.

Shepard, R. N., & Arabie, P. (1979). Additive clustering: representation of similarities as combinations of discrete overlapping properties. *Psychological Review*, *2*, 87-123.

Shepard, R. N., & Chang, J. J. (1963). Stimulus generalization in the learning of classifications. *Journal of Experimental Psychology*, *65*, 94-102.

Shepard, R. N., Hovland, H. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs*, *75*, 13, Whole number 517.

Shepard, R. N., Kilpatric, D. W., & Cunningham, J. P. (1975). The internal representation of numbers. *Cognitive Psychology*, *7*, 82-138.

Sitnikova, T., Rzhetsky, A., & Nei, M. (1995). Interior-branch and bootstrap tests of phylogenetic trees. *Molecular Biology and Evolution*, *12*, 319-333.

Skrondal, A., & Rabe-Hesket, S. (2004). *Generalized Latent Variable Modeling. Multilevel, Longitudinal, and Structural Equations Models.* Boca Raton: Chapman & Hall.

Soli, S. D., Arabie, P., & Carroll, J. D. (1986). Discrete representation of perceptual structure underlying consonant confusions. *Journal of the Acoustical Society of America*, *79*, 826-837.

Stevens, J. (1992). *Applied multivariate statistics for the social sciences*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Stram, D. O., & Lee, J. W. (1994). Variance components testing in the longitudinal mixed effects model. *Biometrics*, *50*, 1171-1177.

Suppes, P., Krantz, D., Luce, R., & Tversky, A. (1989). *Foundations of Measurement, Vol. II, Geometrical, Threshold, and Probabilistic Representations*. New York: Academic Press.

Tajima, F. (1992). Statistical method for estimating the standard errors of branch lengths in a phylogenetic tree reconstructed without assuming equal rates of nucleotide substitution among different lineages. *Molecular Biology and Evolution*, *9*, 168-181.

Takane, Y. (1981). Multidimensional successive categories scaling: a maximum likelihood method. *Psychometrika*, *46*, 9-28.

Takane, Y. (1983). Multidimensional scaling models for reaction times and same-different judgments. *Psychometrika*, *48*, 393-423.

Takane, Y., & Carroll, J. D. (1981). Nonmetric maximum likelihood multidimensional scaling from directional rankings of similarities. *Psychometrika*, *46*, 389-405.

Takane, Y., & Sergent, J. (1983). Multidimensional scaling models for reaction times and same-different judgments. *Psychometrika*, *48*, 393-423.

Tatsuoka, K. K. (1995). Architecture of knowledge structures and cognitive diagnosis: a statistical pattern recognition and classification approach. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (p. 327-359). Hillsdale, NJ: Erlbaum.

Ten Berge, J. M. F., & Kiers, H. A. L. (2005). A comparison of two methods for fitting the INDCLUS model. *Journal of Classification*, *22*, 273-285.

Theil, H. (1971). *Principles of Econometrics*. New York: Wiley.

Tibshirani, R. J. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*, 267-288.

Torgerson, W. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, *17*, 401-419.

Torgerson, W. (1958). *Theory and methods of scaling*. New York: Wiley.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*, 327-352.

Tversky, A., & Gati, I. (1982). Similarity, separability, and the triangle inequality. *Psychological Review*, *89*, 123-154.

Tversky, A., & Hutchinson, J. (1986). Nearest neighbor analysis of psychological spaces. *Psychological Review*, *93*, 3-22.

Verbeemen, T., Storms, G., & Verguts, T. (2004). Similarity and taxonomy in categorization. In K. Forbus, D. Gentner, & T. Regier (Eds.), *Proceedings of the 26th annual conference of the cognitive science society* (p. 1393-1398). Mahwah, NJ: Lawrence Erlbaum.

Verbeemen, T., Vanoverberghe, V., Storms, G., & Ruts, W. (2001). The role of contrast categories in natural language concepts. *Journal of Memory and Language*, *44*, 618-643.

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge UK: Cambridge University Press.

Weinberg, S. L., Carroll, J. D., & Cohen, H. S. (1984). Confidence regions for INDSCAL using the jackknife and bootstrap techniques. *Psychometrika*, *49*, 475-491.

Werner, H. J. (1990). On inequality constrained generalized least squares estimation. *Linear Algebra and Its Applications*, *127*, 379-392.

Werner, H. J., & Yapar, C. (1996). On inequality constrained generalized least squares selections in the general possibly singular gauss-markov model: a projector theoretical approach. *Linear Algebra and Its Applications*, *237*, 359-393.

Winsberg, S., & Ramsay, J. O. (1981). Analysis of pairwise preference data using integrated B-splines. *Psychometrika*, *46*, 171-186.

Wolak, F. A. (1987). An exact test for multiple inequality and equality constraints in the linear regression model. *Journal of the American Statistical Association*, *82*, 782-793.

Wollan, P. C., & Dykstra, R. L. (1987). Algorithm AS 225: Minimizing linear inequality constrained Mahalanobis distances. *Applied Statistics*, *36*, 234-240.

Yang, Y. (2005). Can the strengths of AIC and BIC be shared? A conflict between model identification and regression estimation. *Biometrika*, *92*, 937-950.

Zaki, S., Nosofsky, R., Stanton, R., & Cohen, A. (2003). Prototype and exemplar ac-

counts of category learning and attentional allocation: A reassessment. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *29*, 1160-1173.

Zielman, B., & Heiser, W. J. (1996). Models for asymmetric proximities. *British Journal of Mathematical and Statistical Psychology*, *49*, 127-146.

Zou, H., Hastie, T., & Tibshirani, R. J. (2006). On the degrees of freedom of the Lasso. Submitted manuscript.

Author Index

Subject Index

Summary in Dutch (Samenvatting)

Feature Netwerk Modellen (FNM) zijn grafische modellen die nabijheidsdata met behulp van features weergeven in een discrete ruimte. *Nabijheidsdata* ontstaan wanneer respondenten gevraagd wordt de gelijkenis tussen paren objecten of stimuli te beoordelen op bijvoorbeeld een 5-punts schaal, waarbij een hoge score aangeeft dat een respondent de twee objecten erg op elkaar vindt lijken. Wanneer een groot aantal respondenten dezelfde objectparen hebben beoordeeld, kunnen de scores gebruikt worden om meer inzicht te verkrijgen in de cognitieve processen die een rol spelen bij het onderscheiden van verschillen en overeenkomsten tussen stimuli. In de psychometrie wordt dit type data vaak met meerdimensionale schaaltechnieken (MDS) geanalyseerd. Bij deze technieken worden de objecten afgebeeld als punten in een laag-dimensionale ruimte en is het doel de geobserveerde nabijheidsdata voor de verschillende object paren zo goed mogelijk te benaderen met afstanden tussen de object punten in die ruimte. Er wordt dan aangenomen dat de *psychologische afstand* tussen objecten, in de vorm van nabijheidsdata voortkomend uit de ervaringen van de respondenten, benaderd kan worden met een *metrische afstand* in een laag-dimensionale ruimte.

De assumptie dat een nabijheidsmaat zich als een metrische afstandsfunctie zou gedragen is al in 1977 door onder anderen Tversky in twijfel getrokken. (Zo weten we allemaal dat de beleving van de lengte van dezelfde treinreis met en zonder spannend boek, duidelijk anders is en dat de benadering in kilometers geen goede weergave is van deze twee verschillende belevingen.) Als alternatief voor de metrische representatie van een nabijheidsmaat, stelde hij het Contrast Model voor, waarbij de afstand tussen objecten wordt weergegeven in termen van een verzameling (set) kwalitatieve eigenschappen en introduceerde hij de *features* die als basis dienen voor het model. Een feature is een prominent kenmerk van een object. De representatie van objecten als een set features leidt volgens Tversky tot betekenisvollere psychologische modellen aangezien de features beschouwd kunnen worden als de elementen van de mentale processen die een rol spelen wanneer respondenten gevraagd wordt objecten te vergelijken, en als zodanig afzonderlijk getoetst kunnen worden.

In het Contrast Model wordt de nabijheidsmaat voor twee objecten weergegeven als de som van de features die beide objecten gemeenschappelijk hebben, de *common features*, en van de features die beide objecten onderscheiden, de *distinctive features*. De nabijheidsmaat wordt in het Contrast Model niet door een afstand benaderd maar door een lineaire combinatie van een set theoretische constructen: de common features worden gevormd door de intersectie te nemen van de feature sets die elk

object beschrijven en de distinctive features worden gevormd door het symmetrisch set verschil (de vereniging minus de intersectie) van de twee feature sets.

Sinds de introductie van het Contrast Model, zijn verscheidene modellen ontwikkeld die ofwel het gemeenschappelijk deel van het model modelleren (het common features model, CF), ofwel het distinctieve gedeelte (het distinctive features model, DF), of een combinatie van beide. De inleiding van deze dissertatie (Chapter 1) geeft een overzicht van al deze modellen. Feature Netwerk Modellen (FNM) concentreren zich op het distinctieve gedeelte en onderscheiden zich van de andere modellen doordat zij een grafische representatie van het DF model geven in de vorm van een netwerk. De objecten (stimuli) worden voorgesteld als punten in een netwerk. De afstand tussen de twee objecten is nu de afgelegde afstand langs de lijnstukken in het netwerk die de twee objecten met elkaar verbinden. De beste benadering van de nabijheidsmaat is dan het kortste pad tussen de twee objecten, die als punten zijn gerepresenteerd in het netwerk.

Het gegeven dat de nabijheidsmaat tussen twee objecten benaderd kan worden door de kortste pad afstand in het netwerk tussen beide objecten, komt voort uit het feit dat een simpele optelling van het aantal elementen van het symmetrisch set verschil een maat oplevert die voldoet aan de axioma's van een metriek, zoals aangetoond door Goodman (1951, 1977) en Restle (1959, 1961). Heiser (1998) heeft aangetoond dat deze afstand in termen van het symmetrisch set verschil ook in termen van coördinaten kan worden voorgesteld en noemde deze de *feature distance*. Deze afstand is gelijk aan de city-block afstand, ook wel de *Manhattan*-metriek genoemd, die zijn naam ontleent aan de manier waarop afstanden worden bepaald in een stad met uitsluitend rechthoekig op elkaar staande straten. De afstand van A naar B in een dergelijk stadsplan is de som van de lengte van de afzonderlijke *blocks* die gepasseerd worden. Dit, in tegenstelling tot de meer gangbare Euclidische metriek, die de afstand van A naar B niet in blocks meet, maar via een directe lijn van A naar B, zoals dit op landkaarten gebeurt. In Manhattan is de Euclidische afstand van A naar B alleen in "vogelvlucht" af te leggen.

In het kader van FNM houdt de city-block metriek in dat de waargenomen nabijheidsmaat tussen stimuli wordt bepaald door de som van het symmetrisch setverschil op elk afzonderlijk feature, dat gelijk staat aan een dimensie in de ruimte. De feature afstand is dan gelijk aan een city-block metriek in een ruimte met binaire coördinaten die gevormd wordt door de features (features zijn immers binaire variabelen die aangeven of een object een bepaalde eigenschap wel of niet heeft). Dit specifieke geval van de city-block metriek wordt ook wel de *Hamming* afstand genoemd. Dat de nabijheidsmaat benaderd wordt door de som van de ongelijkheden op iedere afzonderlijke dimensie, is een uniek kenmerk van de city-block afstand die daarom ook een additieve metriek wordt genoemd.

Als psychologisch model is de additieve metriek aannemelijk indien de stimuli verschillen op discrete, niet vermengbare dimensies (features), zoals in 1950 al aangetoond door Attneave. Een voorbeeld van dergelijke stimuli zijn de bloempot data van Tversky en Gati (1982), waarbij de twee niet vermengbare dimensies gevormd worden door type plant en type bloempot (zie hoofdstuk 1). Het waargenomen verschil tussen de stimuli (verschillende typen planten in verschillende typen bloempotten) is dan afhankelijk van het waargenomen verschil in type

bloempot *plus* het waargenomen verschil in type plant. Wanneer een gewogen optelling van de elementen van het symmetrisch setverschil wordt genomen, kan het relatieve belang van elk feature voor de oplossing worden bepaald aan de hand van het bijbehorende gewicht, de *feature discriminability parameter* (Heiser, 1998). Elk feature splitst de objecten in twee klassen en de feature discriminability parameter geeft aan hoe zeer deze twee klassen van elkaar verschillen. Deze parameters kunnen geschat worden met behulp van een verliesfunctie gebaseerd op het kleinste kwadraten criterium.

De bijdrage van dit proefschrift bestaat onder andere uit de introductie van statistische inferentie in FNM en in het algemeen voor modellen die gebaseerd zijn op features, aangezien er tot op heden in dergelijke modellen nauwelijks tot geen aandacht is besteed aan het beoordelen van de stabiliteit van de oplossingen met behulp van bijvoorbeeld standaardfouten en betrouwbaarheidsintervallen voor de parameters. Hoofdstuk 2 gaat in op de vraag hoe de bijdrage van elk feature beoordeeld kan worden op basis van de feature discriminability parameters voor het geval dat de features a priori bekend zijn op basis van theorie of eerder onderzoek. Het netwerk in FNM biedt een grafische representatie van de relaties tussen de objecten in termen van features en zou tegelijkertijd beschouwd kunnen worden als een psychologisch model voor de mentale representatie van de relaties tussen de objecten zoals deze naar voren komen in de geobserveerde nabijheidsmaten. De netwerk representatie zelf is echter niet toereikend om het psychologisch model te toetsen. Voor dit doel dienen de feature discriminability parameters, die aangeven welk feature het meest bijdraagt aan de beschrijving van de nabijheidsmaten.

Wanneer de features beschouwd worden als (binaire) predictoren kunnen FNM gezien worden als een univariaat multipele regressie model met als regressiegewichten de feature discriminability parameters. Het multipele regressie model biedt weliswaar een uitgangspunt voor statistische inferentie, maar de standaard procedures gaan niet op voor de FNM, aangezien er positiviteits restricties gelden voor de feature discriminability parameters: deze stellen namelijk lijnstukken in het netwerk voor en negatieve lijnstukken hebben immers geen betekenis en leveren dus ook geen adequate beschrijving van een psychologische theorie op. Daarom worden in FNM de parameterschattingen verkregen met behulp van het kleinste kwadraten criterium met positiviteitsrestricties, bekend als *nonnegative least squares*.

Data analyse met restricties op de waarden van de parameters is een veel voorkomend probleem in de statistische literatuur, echter, statistische inferentie voor dit soort problemen is niet eenvoudig omdat in veel gevallen geen statistische theorie beschikbaar is. De theorie over standaard fouten bij least squares met positiviteitsrestricties, beschreven in een bijna vergeten artikel door Liew (1976), blijkt goed toepasbaar te zijn in de context van FNM. In hoofdstuk 2 worden in een Monte Carlo studie deze theoretische standaard fouten, die nog niet eerder getoetst waren in de praktijk, vergeleken met empirische standaard fouten verkregen met de bootstrap methode. De resultaten zijn bemoedigend: de theoretische standaard fouten presteren over het algemeen even goed als de empirische standaard fouten, hetgeen betekent dat volstaan kan worden met het berekenen van een theoretische standaard fout in plaats van een meer tijdrovende bootstrap uit te voeren.

De resultaten van hoofdstuk 2 beperken zich tot het geval dat de features van

tevoren bekend zijn, op basis van theorie of eerder onderzoek. Hoofdstuk 3 biedt een uitbreiding van de statistische inferentie theorie op twee onderdelen. De eerste uitbreiding betreft het geval waarin de features niet van tevoren bekend zijn. Ten tweede blijken de behaalde resultaten voor de theoretische standaard fouten in FNM ook toepasbaar op aan FNM verwante modellen, namelijk de in de psychologie veel gebruikte additieve bomen (*additive trees*), die in de biologie en andere gerelateerde wetenschappen bekend staan als phylogenetische bomen (*phylogenetic trees* of *phylogenies*) en die veelvuldig gebruikt worden om genetische verwantschap aan te tonen tussen organismen. Deze boomstructuren zijn een speciaal geval van FNM wanneer de features een bepaalde structuur hebben, namelijk wanneer de set features bestaat uit uitsluitend geneste features en aangevuld met een set interne nodes, die aangeven waar clusters van objecten zich voordoen, of in het geval van de phylogenetische bomen, een afsplitsing in verschillende organismen plaatsvindt.

De resultaten van hoofdstuk 3 laten zien dat de methode om theoretische standaard fouten en bijbehorende 95% betrouwbaarheidsintervallen te berekenen voor de feature discriminability parameters in FNM ook toepasbaar is voor additieve bomen of phylogenies. Waarbij moet worden vermeld dat de positiviteitsrestricties voor de lijnstukken (geschat als de feature discriminability parameters) bij boomstructuren nog veel meer van belang zijn dan bij FNM aangezien elk lijnstuk exact 1 feature voorstelt. De resultaten berperken zich niet tot het geval waarbij de featurestructuur al bekend is, maar gelden ook voor nog niet bekende featurestructuren. In dit laatste geval is een extra stap nodig in het bepalen van de standaardfouten en bijbehorende 95% betrouwbaarheidsgebieden. In een cross-validatie opzet wordt de steekproef opgedeeld in twee sets, een training set om de boomstructuur (feature structuur) te vinden en een test set waarop de gevonden feature structuur gefit wordt om de standaardfouten en de betrouwbaarheidsintervallen te verkrijgen.

Deze resultaten zijn van belang omdat tot op heden in de psychologische literatuur nog geen statistische inferentie is toegepast op boomstructuren. Het phylogenitisch domein kent echter wel een traditie van statistische inferentie. Opvallend is dat in vrijwel geen enkele methode om phylogenies te fitten, gebruik gemaakt wordt van positiviteitsrestricties op de parameters die de lijnstukken voorstellen. Het gebruik in FNM van het multipele regressie raamwerk gecombineerd met features kan een waardevolle aanvulling voor phylogenetische bomen betekenen. Niet alleen kunnen afsplitsingen van verschillende organismen getoetst worden door een feature aan de set toe te voegen, maar de multipele regressie context biedt ook de mogelijkheid eenvoudig een algemene cross-validatie statistiek te bereken waarmee de fit van verschillende boomstructuren systematisch vergeleken kan worden, ook als deze niet genest zijn. Nadeel van de voorgestelde theoretische standaardfouten is dat zij berusten op de assumpties van normaal verdeelde error termen en homogeniteit van de varianties, beide niet altijd aannemelijk in de praktijk van de data analyse.

Hoofdstuk 4 van dit proefschrift bouwt voort op de situatie waarin de features niet a priori bekend zijn en introduceert een methode waarmee een adequate subset features gevonden kan worden op een manier die verwant is aan het predictor selectie probleem in de context van multipele regressiemodellen. De voorgestelde methode begint met het opstellen van de complete set distinctieve features voor

een gegeven aantal objecten. Aangezien features binaire variabelen zijn, kunnen zij eenvoudig gegenereerd worden met binaire codering. Om een aantal practische redenen, is gekozen voor de Gray code, een speciale vorm van de standaard binaire codering waarbij elk opeenvolgende binaire vector slechts op één bit verschilt van de voorafgaande vector. De tweede stap van de methode is de selectie van een subset features uit de totale set met behulp van de *Lasso*, uitgevoerd met het recentelijk ontwikkeld Least Angle Regression (LARS) algoritme (Efron et al., 2004). De Lasso is een predictor selectie methode voor multipele regressie modellen die een subset predictoren (in dit geval features) selecteert op basis van een compromis tussen model fit en model complexiteit. Het uitgangspunt is niet een optimale subset voor de gegeven data, maar een subset features die goede predictieve eigenschappen bezit, dat wil zeggen, een goede fit zal hebben op een nieuwe steekproef.

Voor gebruik met FNM moest de Lasso eerst aangepast worden om aan de positiviteitsrestricties te voeldoen en dit heeft geleid tot de ontwikkeling van de Positieve Lasso. De methode is ook toe te passen op a priori gegeven features en aangezien de beste subset features geselecteerd wordt, kan dit dienen als alternatief voor het kiezen van de features die het meest bijdragen aan de oplossing met behulp van betrouwbaarheidsintervallen. Een nadeel is echter dat de methode alleen goed werkt voor aantallen objecten niet groter dan 22 omdat de te genereren complete set distinctieve features dan 2 miljoen bedraagt en deze in de huidige applicatie niet door de computer bewerkt kan worden.

Naast de verschillende bijdragen op het gebied van statistische inferentie en modelselectie in de hoofdstukken 2 tot en met 4, richt het laatste hoofdstuk van dit proefschrift zich op de netwerk representatie van FNM. Hoofdstuk 5 laat zien dat de netwerk representatie universeel is voor alle city-block modellen. Dit resultaat komt voort uit het feit dat de city-block afstand een additieve metriek is en maakt gebruik van een aantal kernelementen van de netwerk representatie, zoals metrische segment additiviteit, betweenness en de interne nodes. Deze netwerk representatie is niet alleen universeel voor alle city-block modellen gebaseerd op distinctieve features, maar geldt ook voor modellen gebaseerd op common features, zoals additief clusteren, hierarchische boomstructuren en additieve boomstructuren. Een overzicht van de relaties tussen deze modellen is te zien in Figuur 5.10.

Eerder in deze samenvatting werd gemeld dat FNM een *distinctive features* model (DF) is en in zekere zin tegenovergesteld aan het *common features model* (CF). Er is een duidelijke relatie tussen de twee modellen: Sattath en Tversky (1987) en later Carroll en Corter (1995) hebben aangetoond dat het CF model en het DF model in elkaar vertaald kunnen worden. Echter dit theoretisch resultaat was nog niet in de praktijk van data analyse uitgeprobeerd. In hoofdstuk 5 wordt de translatie van CF naar DF toegepast op empirische data. Het blijkt mogelijk te zijn voor elk gefit CF model een even goed fittend DF model te vinden met gebruik making van dezelfde common features en feature gewichten en hetzelfde aantal onafhankelijke parameters. Een belangrijk resultaat dat hieruit volgt is dat een model dat het CF model met het DF model combineert, ook uitgedrukt kan worden als een combinatie van twee afzonderlijke DF modellen, waarmee het DF model een algemener model blijkt te zijn dan het CF model.

Hoofdstuk 6 sluit het proefschrift af met een algemene conclusie en een discussie.

De voor- en nadelen van het gebruik van de theoritsche standaard fouten in FNM worden uiteengezet en vergeleken met de bootstrap standaard fouten, waarbij het vooral gaat om de assumpties van normaliteit, homogeniteit van de varianties en het probleem van alpha inflatie bij het gebruik van betrouwbaarheidsintervallen voor meerdere features. Mogelijke oplossingen en ideeën voor vervolgonderzoek worden aangedragen. Naast de onderwerpen van statistische inferentie en model selectie, worden ook de netwerkrepresentatie, de embedding van het netwerk in een ruimte van lagere dimensionaliteit besproken.

Curriculum vitae

Laurence Emmanuelle Frank werd geboren op 8 september 1969 te Delft. In 1987 behaalde zij het diploma gymnasium A aan het Stedelijk Gymnasium te Schiedam. Hierna volgde de studie Franse Taal- en Letterkunde aan de Universiteit Leiden die in 1992 afgesloten werd met het doctoraal examen, en in 1993 werd de eerstegraadsbevoegdheid Frans aan dezelfde universiteit behaald. Vanaf 1994 volgde de studie Psychologie die in 2001 cum laude werd afgerond in de afstudeerrichting Methoden en Technieken. Van 2001 tot 2005 was zij aangesteld als assistent in opleiding aan de afdeling Methoden en Technieken aan de Universiteit Leiden. Thans is zij als onderzoeker verbonden aan de afdeling Methoden en Technieken van de Universiteit Utrecht waar zij onderzoek doet naar de analyse van randomized response data.