Feature network models for proximity data : statistical inference, model selection, network representations and links with related models

Frank, L.E.

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*	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

*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 **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*.

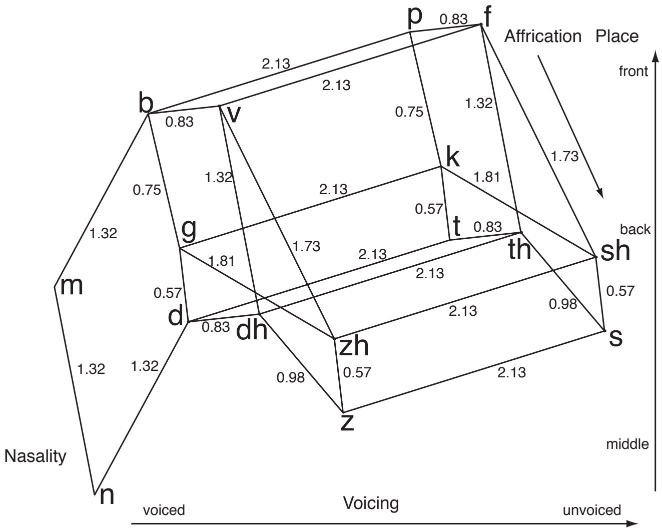

Figure 2.1: Feature Network Model on consonant data (dh = ð; zh = ʒ; th = θ; sh = ʃ).

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 $\boldsymbol{\lambda}_{\mathrm{KT}}$ 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 $\hat{\boldsymbol{\eta}}_{\mathrm{ICLS}}$ is bounded, i.e., all elements satisfy the constraint $-\mathbf{A}\hat{\boldsymbol{\eta}} \leq 0$ (Equation 2.9), then all elements of $\boldsymbol{\lambda}_{\mathrm{KT}}$ 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

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

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

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

where

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

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

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

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 $\hat{\boldsymbol{\epsilon}}$:

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

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 $\hat{e}_l$ 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^a_B	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^a_B	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.

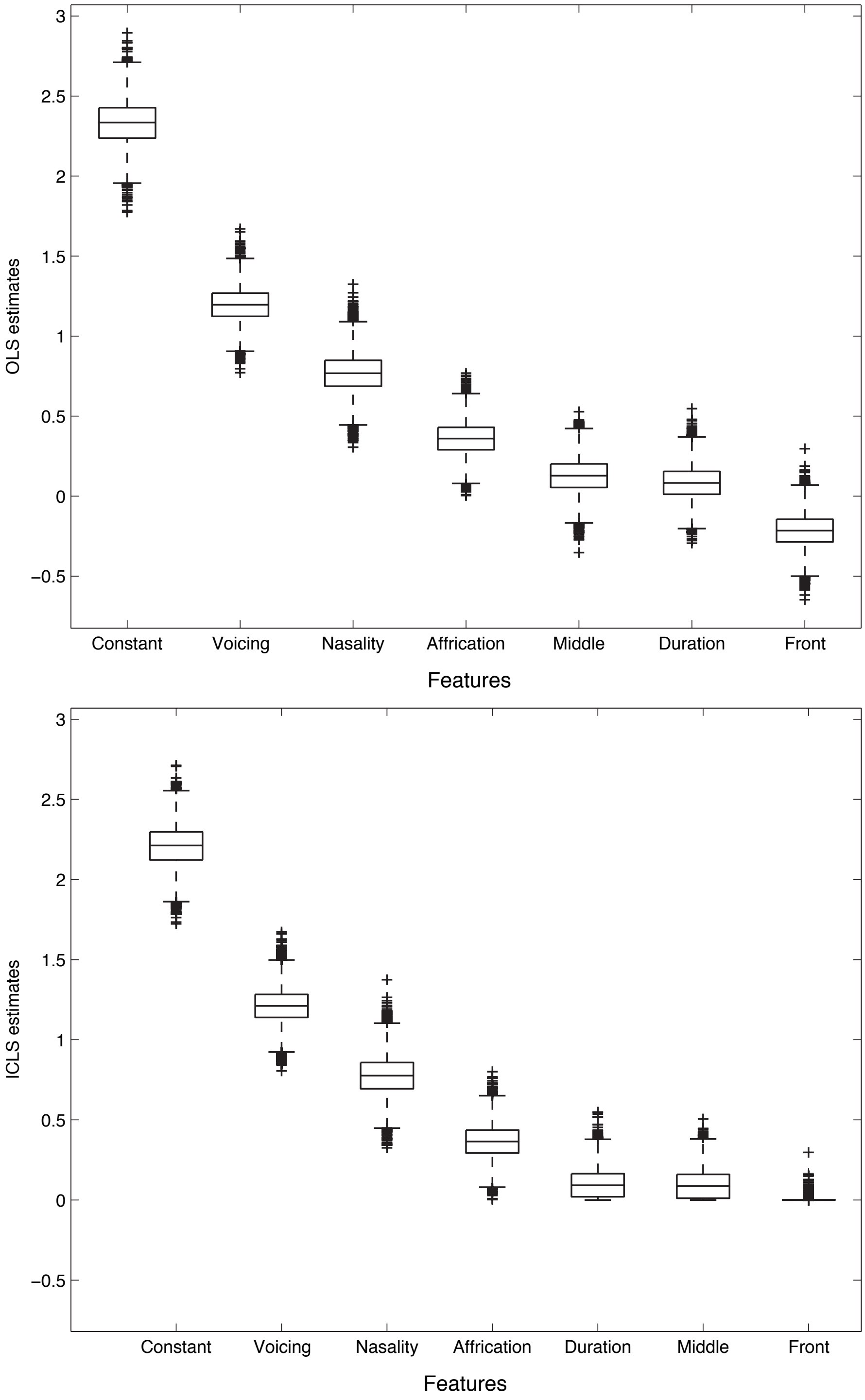

Figure 2.2: Empirical distribution of OLS (top) and ICLS (bottom) estimators (1,0000 bootstrap samples).

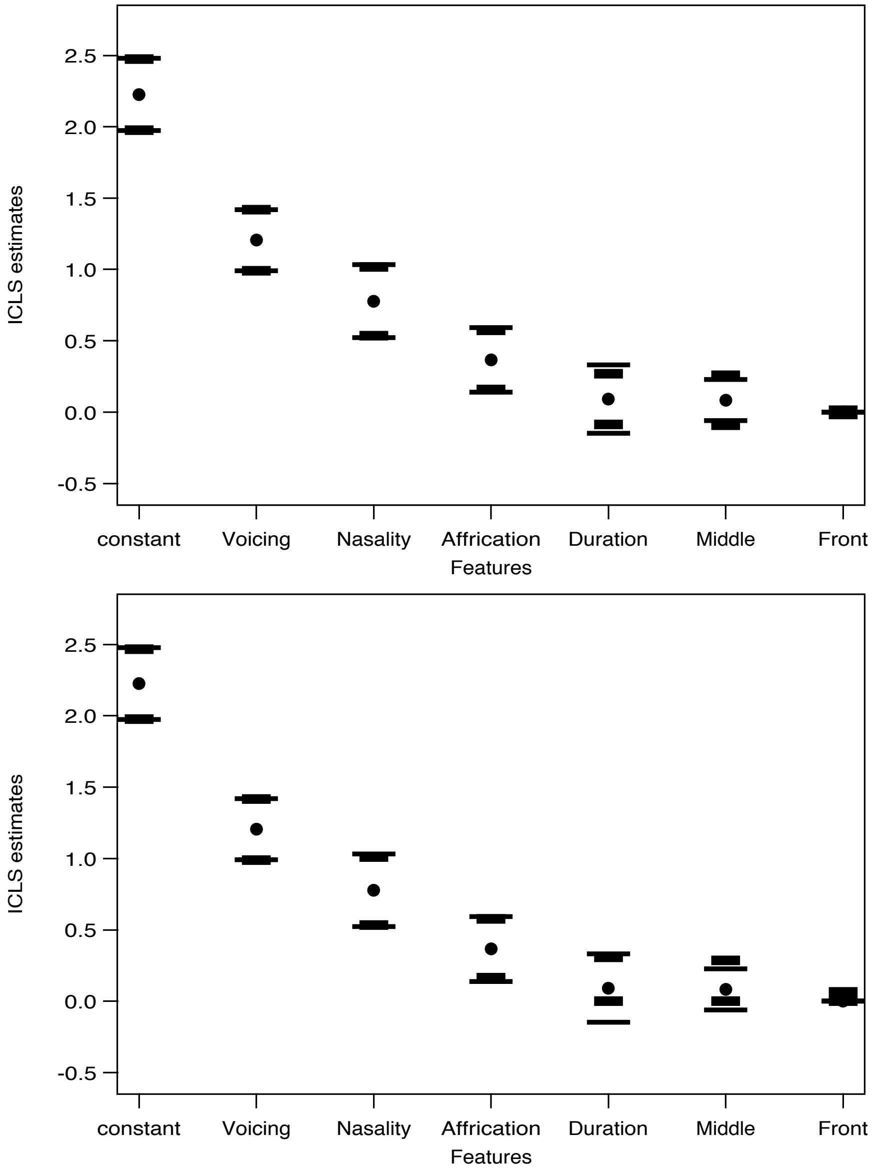

Figure 2.3: Comparison of nominal confidence intervals for ICLS estimator with bootstrap-t CI (top) and bootstrap BC_a CI (bottom); long bar = nominal CI; short bar = bootstrap-t CI or BC_a CI.

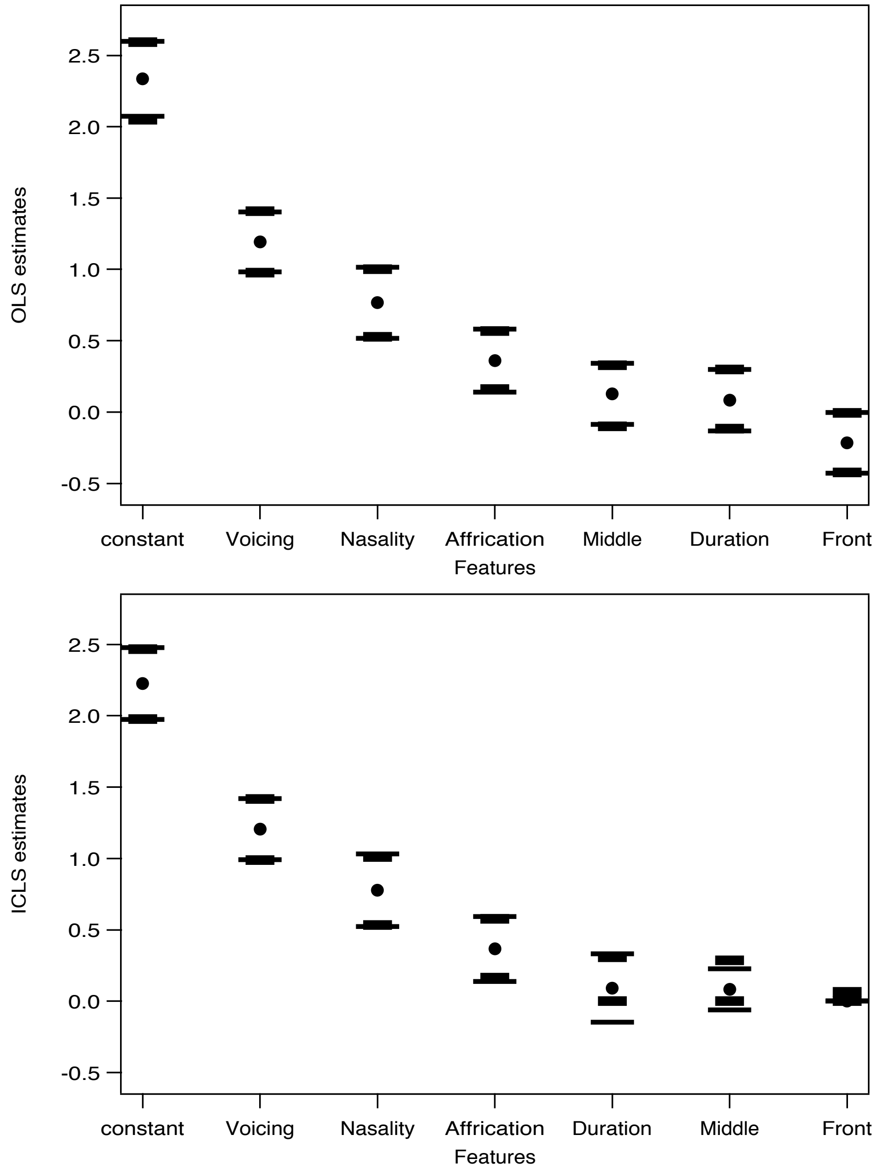

Figure 2.4: BC_a and nominal confidence intervals for OLS and ICLS estimators (long bar = nominal CI; short bar = BC_a CI).

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

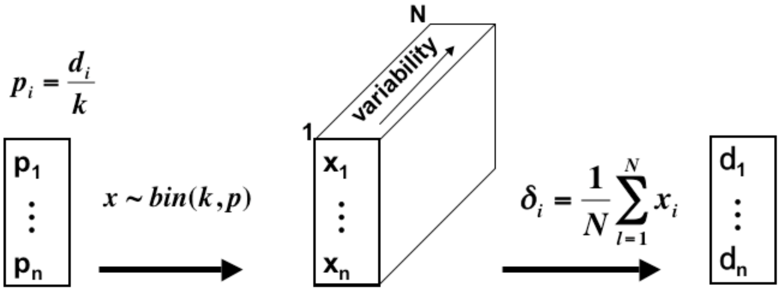

Figure 2.5: Sampling dissimilarities from a binomial distribution

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):

$$rmse = \sqrt{sd^2_{\hat{\theta}} + bias^2_{\hat{\theta}}}. \tag{2.25}$$

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).

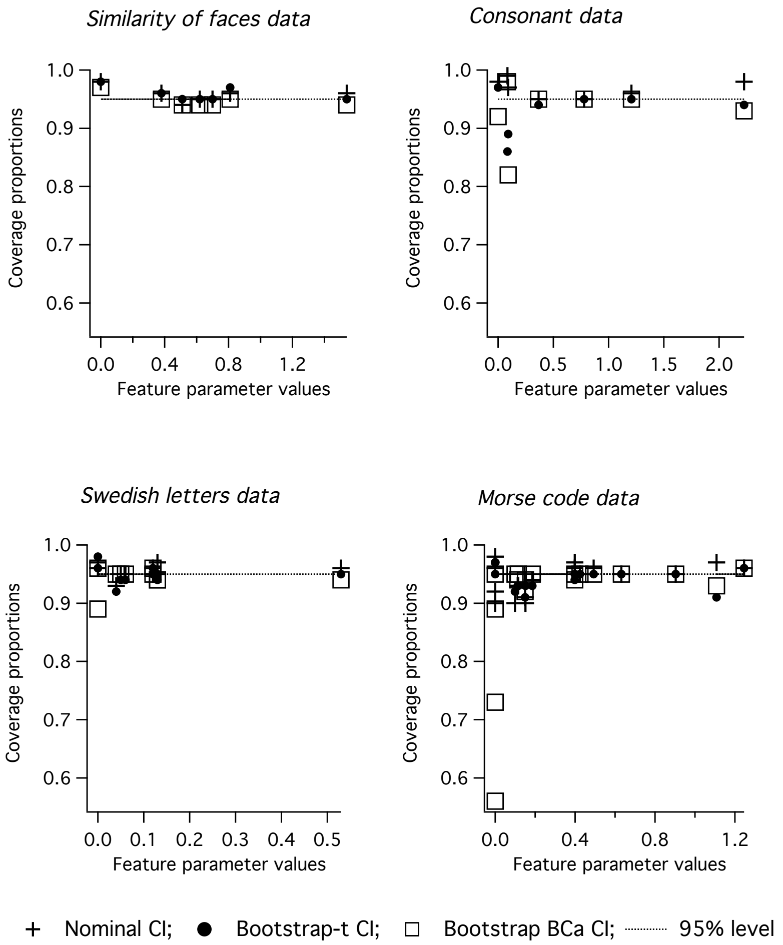

Figure 2.6: Coverage Nominal CI, Bootstrap-t CI, and BC_a CI for ICLS estimates for all simulation studies. The order of the plots follows the increasing number of zero and close to zero parameters present in the data.

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.