Analysis of Ranked Preference Data

Kristine Frisenfeldt Thuesen

Kongens Lyngby 2007
IMM-M.Sc.-2007-50

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-MASTER: ISSN 0909-3192

Preface

This thesis is documentation for a master project, at the department of Informatics and Mathematical Modeling (IMM) at the Technical University of Denmark (DTU).

The project accounts for 30 ECTS points, and the work has been carried out fall 2006 and spring 2007, at the section of Theoretical Statistics, supervised by Professor Per Bruun Brockhoff.

Kgs. Lyngby, May 2007

Kristine Frisenfeldt Thuesen

Abstract

This thesis is addressing theory on models for ranked preference data, based on paired comparisons. The three models,which will be in focus are *Mallows-Bradley-Terry* (MBT), *Bradley-Terry-Luce* (BTL) as presented in [4], and a variation of the MBT model, where it is assumed that the number of times each ranking occurs in the data, is Poisson distributed in stead of polynomial distributed.

The predominant body of the literature on the subject today is limited to a data analytical approach. Either applying the models in a concrete analysis of data, or presenting a theoretic description of how to use the models in the readers forthcoming practical application.

This thesis will in contrast to the existing literature on the subject provide the reader with a thorough description of the models all from the psychophysical idea to the mathematical formulation of the model and the model inference, in a theoretical and strict mathematical way.

The data analytical approach in the existing literature might be the reason why an introductory description of theoretical comparison of the models is missing. This comparison will be made in this thesis.

The attention will especially be on the restraining factors for the ability to describe ranking preference data, and the possibilities for writing them as *Generalized Linear Models* (GLM), to be able to derive Maximum Likelihood estimates through an *Iterative Reweighted Least Squares* (IRLS) algorithm.

For practical approaches a very relevant extension of the ranking models is the

extension to handle panel segmentations.Therefore this work will in the last chapters turn the focus to *Latent Class Models* (LC), which can handle the presence of unobserved segmentation of the consumers. Like other mixture models the inference from the latent class models will happen iteratively by using an Expectation-Maximization (EM) algorithm.

Through the thesis illustrative and relevant tests are made on simple test data, to assist the theoretical descriptions. In a separate chapter some more realistic ranked preference data from the Danish audio- and videoproducer, Bang & Olufsen (B&O) is analyzed.

Resumé

Denne afhandling beskriver teori om modeller for rangordnet preference data, baseret på parrede sammenligninger. De tre modeller, der vil være i fokus er *Mallows-Bradley-Terry* (MBT), *Bradley-Terry-Luce* (BTL) som den præsenteres i [4] og en variant af MBT, hvor det antages at antallet af gange en given rangordning optræder i data, følger en poisson fordeling i stedet for en multinomial fordeling.

Den litteratur, der i dag er til rådighed indenfor emnet, er i høj grad begrænset til at have en dataanalytisk tilgang. Enten med fokus på anvendelse af modellerne i en konkret analyse opgave, eller i en teoretisk presentation af modeller, med fokus på at lette læserens forestående praktiske anvendelse.

Denne afhandling vil derfor skille sig ud fra den eksisterende litteratur på området, ved at beskrive modellerne helt fra de grundlæggende psykofysiske tanker til den endelige matematiske model samt parameter estimation, på en matematisk teoretisk stringent måde.

Den dataanalytiske tilgang i den eksisterende litteratur er måske årsag til en manglende introducerende teoretisk sammenligning af de nævnte modeller, hvilket denne afhandling vil råde bod på.

Fokus vil i særdeleshed ligge på modellernes begrænsninger i forhold til at beskrive rangordnet preference data, samt mulighederne for at opskrive dem som *Generaliserede Lineære Modeller* (GLZ), med henblik på at kunne finde Maksimum Likelihood estimater gennem en *Iterativ Reweighted Least Squares* (IRLS) algoritme.

I praktiske anvendelser er en meget relevant udvidelse af rangordningsmodellerne at kunne håndtere ikke-homogene forbrugergrupper. Derfor vil sidste del af afhandlingen beskrive *latent-klasse modeller* (LCM), der kan håndtere tilstedeværelsen af uobservede segmenteringer af forbrugerne. Som andre mixtur modeller vil fastlæggelsen af paramtrene af latent-klasse modellen ske iterativt ved brug af en *Expectation-Maximization* (EM) algoritme.

Gennem afhandlingen vil relevante, illustrative test blive foretaget på simple datasæt, med det formål at understøtte de teoretiske beskrivelser. I et separat kapitel vil et mere realistisk datasæt fra den Danske højtaler- og tv-producent, Bang & Oluftsen (B&O) blive analyseret.

Acknowledgements

It is no secret that most of my master courses have been concentrated within numerical analysis and optimization. This master project has therefor been a bit on the border of my field, but Per has been very good at translating words and concepts extending my statistical vocabulary.

I'd like to thank Per for this work, and for the meetings we have had, which without exception have refilled my enthusiasm for the project. I find you a very good supervisor.

A would also like to thank Philippe Courcoux, Unité de Sensométrie et Chimiométrie, at ENITIAA for taking time to answer questions on his papers [4] and [6].

Rank data from the Danish company Bang & Olufsen, was kindly provided by Søren Bench, together with the report [14] and information about how the group at B&O had analyzed the data. Thank you.

Contents

List of Symbols

$\mathbf{1}$	vector of ones
$a(\phi)$	function in distribution from the exponential family
α	mode in Gumbel distribution
α_s	prior probability of latent class s
$b(\eta)$	function in distribution from the exponential family
b_0	constant
β	scale parameter in Gumbel distribution
β	parameter vector in GLM
$c(\mathbf{y}, \phi)$	function in pdf from the exponential family
$c(\theta)$	normalization constant
$c_0(\theta)$	normalization constant
pdf	cumulative distribution function
D	deviance
D_c	Aikaikes Information Criterion
$E(\cdot)$	expected value
η	canonical location parameter in f_{exp}
ε	Eulers constant
f_{bin}	binomial pdf
f_{exp}	pdf from exponential family
f_{Poiss}	Poisson pdf
$g(\cdot)$	link function

GLM	generalized linear model
h_u	ordering number u
i, j	indexs of items
k	index of consumers or number of latent classes
$L(\cdot)$	likelihood function
$\ell(\cdot)$	log-likelihood function
λ	mean in Poisson distribution
$\log(\cdot)$	natural logarithm
$\mathrm{LOG}(\cdot)$	pdf of logistic distribution
μ	mean
n	number of consumers
p	probability
p_u	probability of $Y_{uk} = 1$
$P(\cdot)$	probability of event
p_{ij}	probability of $Y_{ij} = 1$
π_i	item parameter
$\Phi(\cdot)$	Gaussian cdf
$\Phi(\cdot)^{-1}$	probit function
ϕ	dispersion parameter in f_{exp}
pdf	probability density function
r_u	ranking number u
r_{iu}	rank value of item i in rank r_u
R_u	event that $Y_{uk} = 1$
s	index of latent classes
σ	standard deviation of Normal distribution
t	number of items
θ_i	item parameter, scale value of item
$\boldsymbol{\theta}$	vector of item parameters
Θ	matrix with rows as item parameter vectors
ϑ	normalization parameter
U_i	uniform distributed s.v.
W_i	s.v. of item preference
X	model matrix
$\mathbf{y}$	observation of $\mathbf{Y}$
y_{ijk}	observation of Y_{ij} from consumer k

$\mathbf{Y}$	multiple s.v. of the data observed
Y_{ij}	s.v. of preference between item i and j
Y_u	s.v. sum over consumers of Y_{uk}
Y_{uk}	s.v. indicating ranking preference of consumer k

Chapter 1

Introduction

When approaching literature on analysis of ranking data, it is hard to find literature giving a precise mathematical, but introductive presentation of the subject.

Most of the literature found will be very desciptive on the experiments or maybe on how to use some specific method, what is very seldom is literature with the purpose of introducing the theory in a mathematical way.

This thesis will aim to describe both the motivation for, the construction and use of *paired comparison based ranking models.*

Motivation for Preference Ranking Models

A person has to decide which pair of trousers in a shop to buy. Each pair of trousers have some different options such as color, number of pockets, zipper or buttons etc, and the person has a preference of each pair build upon all these options.

The person might start by sorting the trousers into categrories, red trouses with pockets in one box, short green trousers in one box and etc. If a whole group of consumers were asked to sort the same set of trousers, in the same boxes, cathegorical data would occour. This data could be analysed, but it would not tell anything about the preference of the consumers.

If all consumers were assumed to have the same preference for trousers, and each consumer was asked to rank the trousers in the shop so that the pair he like the most is on the hanger in front and the pair he like the least is far behind, the information about relative preference, would suddently be a part of the data. By numbering the hangers, numerical data occure.

The same ranking would be expected except for minor differences, due to randomness. Eg. one person didn't notice the pockets on the back, one person didn't notice that he switched two pairs, when he hurried hanging up the trousers etc.

An intuitive way of modeling the preference from the ranking observations, would be to analyze the number on the hangers through ordinary regression analysis. This analysis would estimate the most likely ranking, but some information in the rank data would be lost.

If two trousers are alike except for a minor difference in the colors, they would be likely to hang next to each other in the rankings. The more alike, the more persons might not notice the difference between them and therefor the ranking of those two pairs will often be switched.

This information about the distance between the trousers on the preference scale would not be found by regression on the hanger numbers. Therefor models incorporating this information is needed if the relative difference in preference is to be estimated. Such models are called ranking models, and can be thought of as both scaling and regression. Therefore not only the correct order of the trousers is estimated using ranking models, also the relative distance between the hangers is found.

Through the years a lot of different approaches has been put to estimate preference from rank data. In section 5 different kind of ranking models are briefly introduced, such as Proportional Odds models, Distance based ranking models and Multistage ranking models.

The main focus of this thesis is however put on ranking models based on paired comparisons.

Structure of the Thesis

The structure of the thesis is as follows:

The inference of the models in this thesis is done by maximum likelihood estimation. In order to derive the estimates through an Iterative Reweighted Least Squares algorithm, theory on Generalized Linear Models (GLM) is presented in section 2.

The origin of preference data and the ideas behind the first attempts to model preference or attitude is described in section 3. In this section a data set for

In section 4 paired comparison (PC) models will be presented, as an introduction to the PC based ranking models, which account for the main focus of this thesis.

A brief overview of some ranking models is given in section 5, to present the diversity in which the PC based ranking models are found. Section 6 present the PC based ranking models.

Models coping with panel segmentations are presented in section 7 under the name Latent Class Ranking Models.

In section 8, rank data from B&O is modeled using PC based models in section, through multivariate statistical analysis on the estimates is carried out, together with an analysis of the panel segmentation in the data.

A conclusion will summarize, emphasizing the main results of the project, in section 9.

Chapter 2

Generalized Linear Model

From one research field to the other the word ”model” might be used for a lot of different concepts. In the following section the meaning of the word *model* within this *thesis* will be emphasized, followed by a description of a specific kind of models, Generalized Linear Models (GLM).

2.1 What is a mathematical model?

To be able to make calculations on physical problems, a mathematical model of the problem is needed. A mathematical model is always formed as a compromise between the complexity of the physical problem, the accuracy needed and the capability of performing function evaluations.

Data is the concrete link to the physical problem, and provide information from the physical problem. The model describe the overall origin of the data in a mathematical formulation. The model is the structure, with a number of free variables.

There are different ways of approaching the choice of model or density function as it is called within the field of statistics. According to [2], they are often divided into parametric and non-parametric models.

The definitions are ambiguous, but as a guideline, parametric models can be thought of as models assuming density function from a specific kind of distributions. Data is then used to estimate the parameters, eg. mode and scale parameters, to define the best member of the family of variables.

The non-parametric models on the other hand can be thought of as models where the data is used to "design" the density function, allowing for much more general models than the parametric. Such models will fit data well, but a drawback is that the number of variables will often increase by the size of the data. The choice of how flexible the model must be can only be made from a decision of how smooth the solution shall be.

Until section 7 the models presented will be parametric. In section 7, a semi-parametric model or mixture model will be presented. A model, which is a mixture of parametric models.

When the model structure has been decided, the question comes to how to determine the parameters, that is how to tie free parameters of the model. This is often done by minimizing some error function. This step is called inference of the model. Inference of a GLM model is mentioned in section 2.5.

Depending on the choice of inference, an algorithm for doing the actual derivations of the parameter estimates, must be found. And finally a choice of a specific implementation of the algorithm, must be made.

In some cases the minimization of the error function demand a solution to a non linear problem. In such cases the algorithm must be iterative. This is also the case when inferencing from GLMs.

A commonly used model, is the general linear model, which can be defined as a model of the vector of responses $\mathbf{Y} = (Y_1, \ldots, Y_q)$, where the components Y_i are mutually independent normally distributed with common variance. This entail that the model can be written as

$$\mathbf{Y} = X\beta + \epsilon,$$

where the linear predictor part is defined $X\beta$, and ϵ is normal distributed.

Once the model is formed as a general linear model a lot of statistic packages have standard procedures for evaluating the model in many ways. (regression analysis, analysis of variance, analysis of covariance etc.) But of cause not all observations can be modeled by general linear models.

2.2 What is a Generalized Linear Model?

Generalized linear models are generalizations of general linear models, where the mutually independent components of $\mathbf{Y}$ are allowed to be distributed according to the exponential family of distributions.

The advantages of describing models within the GLM framework is that a lot of the standard procedures for evaluating the model is inherited from the general linear models.

The reason for restricting GLMs to the exponential family of distributions for $\mathbf{Y}$ is that the same algorithm applies to this entire family, for any choice of link function, [1, p. 119].

2.3 The Exponential Family of Distributions

The definition of the exponential family distribution vary over the literature, in this thesis the most general will be used, following [16], defining a pdf for a distribution from the exponential family as

$$f_{\exp}(y;\eta,\phi) \quad = \quad \exp\left(\frac{(y\eta - b(\eta))}{a(\phi)} + c(y,\phi)\right), \tag{2.1}$$

where η is a function of the location parameter called the canonical parameter and ϕ is the dispersion parameter.

Different members of the exponential family of distribution occur for different definitions of the functions a, b and c. Examples of members could be the Normal, the exponential, the binomial, the Poisson and the Gamma distribution.

The mean of $\mathbf{Y}$ can be found as,

$$E(\mathbf{Y}) = b'(\eta), \tag{2.2}$$

and the variance

$$var(\mathbf{Y}) = a(\phi)b''(\eta). \tag{2.3}$$

For derivation of both (2.2) and (2.3) see [16, page 38-39]. Notice that the variance might be a function of the mean, contrary to the general linear model.

2.4 Link Function

A consequence of $\mathbf{Y} = (Y_i, \ldots, Y_q)$ being modeled by a GLM, is that for each component Y_i, with a corresponding set of X_{ij}, there exists a monotone differentiable function g called a link function, relating the expected value, $E(Y_i) = \mu_i$ to a linear predictor so that

$$g(\mu_i) = X_i\beta, \tag{2.4}$$

where $X_i = (X_{i1}, \ldots, X_{iq})$ and $\beta = (\beta_1, \ldots, \beta_v)$.

Every distribution has a link function which in some sense is "natural", called the canonical link function. According to [16, page. 42], the definition of the canonical link function, is the link function, for which $g(\mu) = \eta$, where η is the canonical location parameter of the distribution.

Summing up a generalized linear model is defined by the choice of the distribution of the components of $\mathbf{Y}$, through the functions a,b and c, and by a link function g with corresponding linear predictor $X\beta$.

GLM for count data

When the components of $\mathbf{Y}$ are independent and distributed by a Poisson distribution, the pdf is

$$\begin{aligned} f_{\text{Poiss}}(y_i, \lambda) &= e^{-\lambda}\frac{\lambda^{y_i}}{y_i!} \\ &= \exp(y_i\eta - \exp(\eta) - \log(y_i!)), \end{aligned}$$

where the last equation is recognized as an exponential density function with, $\eta = \log(\lambda)$, $b(\eta) = \exp(\eta)$, $c(y_i, \phi) = -\log(y_i!)$ and $a(\phi) = 1$.

From (2.5) the canonical link function for the Poisson distribution is determined as the logarithm, since $\eta = \log(\lambda)$. Notice that $g : [0;\infty[\curvearrowright] - \infty;\infty[$, spreading the positive count observations to $] - \infty;\infty[$.

GLM for binary and binomial data

When the observations can be thought of as sums of Bernoulli outcomes, the components of $\mathbf{Y}$ might be modeled as binomial distributed, and then the pdf is

$$\begin{aligned} f_{\text{bin}}(y_i, p, n) &= \binom{n}{y_i} p^{y_i}(1-p)^{n-y_i} \\ &= \exp\left(y_i\eta + n\log\left(\frac{1}{1+\exp(\eta)}\right) + \log\binom{n}{y_i}\right), \end{aligned}$$

where the last equation is recognized as a density function from the exponential family, $\eta = \log(\frac{p}{1-p})$, $b(\eta) = n\log(1+\exp(\eta))$, $c(y_i, \phi) = \log\binom{n}{y_i}$ and $a(\phi) = 1$.

According to the definition in [16], mentioned above, the canonical link function of the binomial distribution, should be the function $g(\mu) = \log(\frac{\mu}{n-\mu})$, since it was recognized that

$$\begin{aligned} \eta &= \log\left(\frac{p}{1-p}\right) \\ &= \log\left(\frac{\mu}{n-\mu}\right). \end{aligned} \tag{2.5}$$

However, it is worth noticing, that everywhere in the GLM literature ([16], [1] etc), the canonical link function for the binomial distribution is declared to be the logit function, which according to (2.5), is *not* the case, except for the special case of Bernoulli distribution, where $n = 1$.

In this thesis the canonical link function for the binomial distributed will be defined as (2.5), and referred to as the *scaled* logit or simply the canonical link function for the binomial distribution.

Another link function that is often used instead of the scaled logit, is a *scaled* probit function

$$\begin{aligned} g(\mu) &= \Phi^{-1}(p), \\ &= \Phi^{-1}(\frac{\mu}{n}), \end{aligned} \tag{2.6}$$

where Φ is the cumulative distribution function for the Gaussian distribution. Yet another link function mentioned will be the *scaled* complementary log-log link

$$\begin{aligned} g(\mu) &= \log(-\log(1-p)) \\ &= \log(-\log(1-\frac{\mu}{n})) \end{aligned} \tag{2.7}$$

As for (2.5), the functions given by (2.6) and (2.7), is in the literature misleadingly called relatively the probit link and the complementary log-log link.

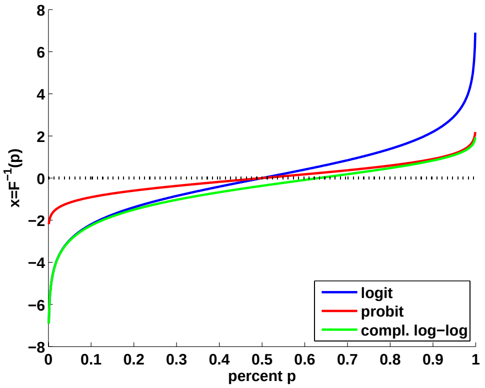


Figure 2.1: Logit, probit and complementary log-log functions.

In Figure 2.1 the logit, probit and complementary log-log functions are compared. Notice that all of them transform goes from $[0; 1]$ to $]-\infty; \infty[$.

2.5 Inference for GLMs

When the kind of distribution or the structure of the model has been decided, the observations should be used for model inference, which means to determine the parameters of the model, so that it is the best in "some sense". In this situation the question raise: What is the definition of "best sense"? Or in other words; how should the degrees of freedom of the chosen structure be tied? The question is often answered by minimizing some distance function or norm of the residuals.

One definition of *best* could be the model that makes the observed data most likely. This estimate of the parameters is called the Maximum Likelihood estimate and is the model that minimizes the variance. In the case of general linear models this is the same as minimizing the 2-norm of the residuals.

Other ways to determine the parameters could be to minimize other distance functions, like the 1-norm and the ∞-norm. Each having different advantages and disadvantages. Eg. if robustness is important, one should chose to minimize the 1-norm, or maybe a combination of the 1- and 2-norm defined as the Huber-function. See [19, chapter 8] for a good description of how the ML estimate relates to other estimates, for a more numerical version of parameter estimation choose e.g. [12].

Anyway in this thesis the definition of ”best sense” will be the parameters maximizing the likelihood of the observations.

2.5.1 Likelihood Equations

Recalling that all GLMs have distributions which are members of the exponential family, maximizing the likelihood is defined as maximizing the probability density function given in (2.1), (in this situation called the likelihood function) with respect to the parameters η and ϕ. Since maximizing the likelihood and the logarithm result in the same estimates, then for convenience maximizing the logarithm of the likelihood is most often chosen.

The log-likelihood function for (2.1) is written as

$$\begin{aligned} \ell(\eta,\phi;y) &= \log(L(\eta,\phi;y)) \\ &= \frac{y\eta - b(\eta)}{a(\phi)} + c(y,\phi). \end{aligned} \tag{2.8}$$

Finding the maximum can be done by finding roots in the derivatives of (2.8). The derivative for a single observation, x_j is

$$\begin{aligned} \frac{\delta\ell}{\delta\beta_j} &= \frac{(y-\mu)}{a(\phi)}\frac{1}{V}\frac{d\mu}{d\eta}x_j \\ &= \frac{W}{a(\phi)}(y-\mu)\frac{d\eta}{d\mu}x_j, \end{aligned}$$

where the weight W is given by

$$W^{-1} = \left(\frac{d\eta}{d\mu}\right)^2 V. \tag{2.9}$$

By summing over all the observations the likelihood equation for one parameter β_j is given by

$$\sum_{i=1}^{n} \frac{W_i(yi-\mu_i)}{a(\phi)}\frac{d\eta_i}{d\mu_i}x_{ij} = 0,$$

and defines one row in a system of equations, which can be solved with respect to β, through the link functions. See [1] for more details.

2.6 Iterative Algorithms

The likelihood equations (2.10) are usually nonlinear in the parameters. And therefore iterative methods must be used to estimate the parameters. There are a lot of different methods, which could come handy. In [1, chapter 4.6] three methods are mentioned, the Newton-Raphson, the Fischer scoring and the Iterative Reweighted Least Squares (IRLS) Method. [16, chapter 2.7] only briefly mention the IRLS.

Newton-Raphson Method

The Newton-Raphson Method is maybe the most common used algorithm for nonlinear optimization, it is a steepest descend method, which means that each step is taken in a direction where the objective function is descending or at least non-increasing. A description using the GLM notation could be found in [1], but every introductory course in optimization has a description. A very good one is found in [13], together with different improved variants of the algorithm.

The Newton-Raphson algorithm among others, uses information about the gradient of the objective function to define descending directions. Methods not using gradient information are also available. Such a *direct search* algorithm is implemented in the software MatLab in the function `fminsearch`. This function will be used in later sections to minimize the negative log-likelihood, in the case where the model can not be formulated as a GLM.

When the GLM framework can be applied other iterative methods are more useful, such as the IRLS method.

Iterative Reweighted Least Squares Method

If the maximization algorithm could take advantage of the structure of the nonlinear objective function, even more effective algorithms would raise. Therefor a lot of algorithms specified for non-linear *least squares* problems have been developed over the years. For an overview of some see [13].

A very used method within the GLM framework is the Iterative Reweighted Least Squares (IRLS) method, where each iteration solves the weighted least squares problem:

$$\begin{aligned} \beta_{new} &= \text{argmin}_\beta\{||W(y - X\beta)||_2\} \\ &= \text{argmin}_\beta\{(y - X\beta)^\intercal W(y - X\beta)\}, \end{aligned} \tag{2.10}$$

where the weighting function $W(\cdot)$ in the case of maximizing the log-likelihood in (2.8) should be given by (2.9). It is stressed that this is only one way to define the weights. The weight function should be chosen according to the problem to be solved. The choice of weighting function can e.g. be used to change the estimate towards a 1-norm estimate.

The IRLS algorithm described by (2.10), is implemented in the statistical software R as the method `glm()`. This implementation will be used in later sections for inference of GLMs.

Chapter 3

Preference Data

One thing is to measure physical merits such as hight, length, weight etc. Another thing is to measure abstract concepts such as preference, attitude, self esteem and intelligence.

In the 19'th century the view of psychology changed. With the development of the subdiscipline *psychophysics*, it was now looked upon as a regular field of science. Psychophysics is dealing with the relationship between physical stimuli and their subjective percepts.

One of the main topics was to find a way to convert the measurable physics to an perceptual experienced scale. This transformation is called *scaling*. A well established scaling is the perceptual experience of the frequency of sound which is called *pitch*, and is measured in *mels*.

3.1 Scaling

When the question comes to scaling more abstract psychological measurements such as attitude, personality traits etc. another kind of research field comes into account. Psychometrics is the research field of theory and techniques of psychological measurements.

The observed data is often answers to a number of questions, and different scaling methods, such as Thurstone scaling, Likert scaling and Guttman scaling, have been developed. A very good text about general issues in scaling is found in [20], which also describe in details the practical procedures of the different scaling methods.

In the following a brief introduction to Thurstone and Guttman scaling will be given together with a mathematical motivation for the scalings. Notice that an assumption for all these scalings is that the abstract concept can be measured on a unidimensional scale.

Thurstone Scaling

In 1928 Thurstone presented the first formal scaling method of measuring attitude in [17], as he was measuring attitude towards religion. He had a number of statements, each which had been applied a numerical merit (a scale value), of how favorable the statement was towards religion. A very descriptive text of the process is given in [20].

The test person could now either agree or disagree with the statements. Thurstone measured the persons attitude towards religion by summing over all the scale values of the agreed statements. This way of measuring attitude is called Thurstone Scaling, and in further analysis of the attitude data, it is assumed to follow a Normal distribution.

As it is a formulated goal of this thesis to present the theory in a mathematical stringent way, a mathematical motivation for the assumed distribution is given. It is stressed that the mathematical motivations are not proved, but are developed by the author as examples of mathematical formulations of the theory. It will however be seen in later sections, that they come to fit nicely into the general theory of Paired Comparison models.

From a mathematical point of view Thurstone assumed that every answer was an outcome of a stochastic variable, weighted by the scale values. The attitude could also be seen as a stochastic variable X, defined as the sum of all the stochastic answer-variables.

A natural choice of distribution of X would be the Normal distribution since the Normal distribution, according to the Central Limit Theorem, can be obtained as the limit distribution of the sum of a large number of independent distributed stochastic variables.

Theorem 3.1 Central Limit Theorem *Let $X_1, X_2, \ldots X_n$ be independent, identical distributed stochastic variables with mean μ and variance σ^2. Then the distribution of*

$$U_n = \frac{\sqrt{n}}{\sigma}\left(\frac{X_1 + \ldots X_n}{n} - \mu\right),$$

will converge towards a normal distribution, when $n \to \infty$, like

$$P(U_n \leq u) \to \Phi(u).$$

Guttmann Scaling

In 1944 Louis Guttman published his work about *Scalogram Analysis*. The Guttman scale is based on observations of a test persons agreement/not agreement answers towards statements just as the Thurstone scale.

Different from the Thurstone scaling a Guttman scale has a cumulative order of the statements. Each statement is applied a value, so that every test person agreeing with statement 4 also agrees with statement 1, 2 and 3. The persons attitude towards say immigration as in the example below, is then measured as the maximum value of the agreed statements.

An example of a Guttman scale could be the Bogardus Social Distance Scale:

1. I believe that this country should allow more immigrants in.
2. I would be comfortable with new immigrants moving into my community.
3. It would be fine with me if new immigrants moved onto my block.
4. I would be comfortable if a new immigrant moved next door to me.
5. I would be comfortable if my child dated a new immigrant.
6. I would permit a child of mine to marry an immigrant.

As for the Thurstone measure of attitude, a more mathematical motivation could be, that every answer of agreement is seen as a stochastic variable. Now if the measure of attitude is defined by the stochastic variable X, then X will follow the maximum distribution of all the stochastic answer-variables.

From the Extremal Types Theorem, a natural choice of distribution for X would be the *Extreme Value Distribution*, also known as the *Gumbel* distribution.

Theorem 3.2 Extremal Type Theorem *For samples taken from a well behaving arbitrary distribution X, the resulting extreme value distribution $Y_n = \max\{X_1, X_2, \ldots, X_n\}$, can be approximated and parameterize with the extreme value distribution (Gumbel) with the appropriate support.*

$$\lim_{n\to\infty} P\left(\frac{Y_n - b_n}{a_n} \leq x\right) = G(x),$$

where $F(x)$ is the Gumbel distribution.

The Gumbel distribution has probability density function as

$$f(x) \quad = \quad \frac{1}{\beta} e^{-\frac{x-\alpha}{\beta}} e^{-e^{-\frac{x-\alpha}{\beta}}}, \tag{3.1}$$

where α is the mode or location parameter and β is the scale parameter.

The mean is derived as $\alpha + \varepsilon\beta$, where the $\varepsilon \approx 0.5772$ is the Eulers number. The standard deviation is $\frac{\beta\pi}{\sqrt{6}}$. The case where $\alpha = 0$ and $\beta = 1$ is called the standard Gumbel distribution.

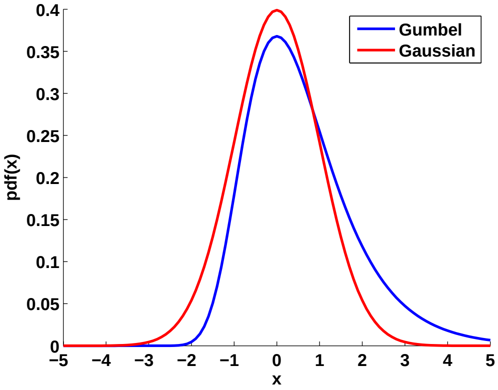


Figure 3.1: Gaussian and Gumbel probability density functions.

Figure 3.1 compares pdf's for the Gaussian distribution with the standard Gumbel distribution. Notice that the Gumbel distribution is not symmetric around the mode.

Item Response Theory

The main body of theory within psychometrics is the *Item Response* (IR) theory. This is theory about the IR models, also known as *Latent Trait Models*, which are mathematical models applying scaling data, used to analyze psychological measurements.

The two IR models which will be used in this thesis are the Thurstone model for paired comparisons, and the Bradley-Terry model. The latter is a special case of the Rash model for dichotomous data. The two models are based on respectively Thurstone scaling and Guttman scaling, and will be presented in section 4.

Chapter 4

Paired Comparison Models

As mentioned in the last section, this section will present theory of the item response models, known as the Thurstone-Mosteller (TM) model and the Bradley-Terry (BT) model. The models specify the probability of a specific outcome in terms of some item parameters, or scale values of the perceived ”weight” on a continuum.

These models are as the title indicates, models for data originating from paired comparison experiments, and are simple examples of proportional odds models also known as order statistic ranking models. In section 5 a generalization of these models into comparison of more than two items is presented. However the real reason for describing the models so detailed in this section is that they are used as a base for the PC based ranking models described in section 6, which are the main-focus models of this thesis.

The section will show why in practise the differences between the Thurstone-Mosteller and the Bradley-Terry model are almost non-existing, but also emphasize that there is a mathematical theoretical interesting difference in how the models have emerged.

4.1 Paired Comparison Data

Assuming that the scale values called *item parameters* of t items are to be estimated. The preferences of a panel of n consumers are used as observations. The consumers are each asked to compare two items at a time, answering which item they prefer of the two.

The consumers are assumed to have equal preference scale, so that the observations can be treated as independent outcomes of the same distribution.

Letting Y_{ij} be a stochastic variable describing the preference between two items, so that

$$Y_{ij} = \begin{cases} 1 & \text{if a consumer prefer item } i \text{ to item } j \\ 0 & \text{otherwize,} \end{cases}$$

for all $i, j = 1, \ldots, t$.

Letting $p_{ij} = P(Y_{ij} = 1)$ be the probability of a consumer preferring item i to item j, the distribution of Y_{ij} is then given as

$$Y_{ij} \quad \sim \quad \text{bin}(p_{ij}, 1) \qquad \forall\ i, j = 1, \ldots, t. \tag{4.1}$$

The preference probability p_{ij} could be thought of as the difference in the consumers preference/attitude towards two items. Therefore the scaling theory from section 3 can be applied. Defining the stochastic variable of the scale value for each item W_i, $i = 1, \ldots, t$, the probability p_{ij} can be modeled as the probability of the event that the preference of item i is greater than the preference of item j,

$$\begin{aligned} P(Y_{ij} = 1) &= P(W_i > W_j) \\ &= P(W_j - W_i \leq 0). \end{aligned} \tag{4.2}$$

So far the theories for the two models, TM and BT agree, but as they are developed from different scaling models the distributions of the latent variables W_i, for $i = 1, \ldots, t$ are different for the two models. Recall that the Thurstone scale assumes Normal distribution of W_i, while the Guttman scale assumes Gumbel.

4.2 Thurstone-Mosteller model

Assuming that the mathematical formulation of the Thurstone scaling is correct, the latent item variables W_i are assumed to follow a Normal distribution with means, μ_i and for simplicity equal variance σ^2,

$$W_i \sim N(\mu_i, \sigma), \qquad \forall i = 1, \ldots, t.$$

The preference probability defined by (4.2), can now be modeled as

$$\begin{aligned}
p_{ij} &= P(W_j - W_i \leq 0) \\
&= P\left(\frac{(W_j - W_i) - (\mu_j - \mu_i)}{\sqrt{2\sigma^2}} \leq -\frac{\mu_j - \mu_i}{\sqrt{2\sigma^2}}\right) \\
&= \Phi\left(\frac{\mu_j}{\sqrt{2}\sigma} - \frac{\mu_i}{\sqrt{2}\sigma}\right) \\
&= \Phi(\theta_j - \theta_i), \qquad (4.3)
\end{aligned}$$

where the last equality sign comes from defining the parameter θ_i as

$$\theta_i = \frac{\mu_i}{\sqrt{2}\sigma} \qquad \forall i = 1, \ldots, t,$$

without loss of model complexity.

This result is consistent with the *Law of Comparative Judgement* (LCJ). In 1927 Louis Leon Thurstone published a paper on the so called ”law”, which is more correctly to be described as a measurement model. The ”law” shows how to model scale values of the perceived ”weight” on a continuum by use of paired comparisons. The fifth case of LCJ, assumes uncorrelated observations and could therefore be used in the present case.

4.3 Bradley-Terry model

Using the Guttman Scaling theory the latent variable W_i of the item i, should follow a Gumbel distribution. This assumption results in the formulation known as the Bradley-Terry probability, as a competitor to (4.3)

$$p_{ij} = \frac{\pi_i}{\pi_i + \pi_j},$$

where parameter $\pi_i \geq 0$ and $\pi_j \geq 0$ for $i, j = 1, \ldots, t$. The following section will provide the argumentation.

Assume that the observed response Y_{ij} is formed by a latent variable. Let W_i for $i = 1, \ldots, t$ be stochastic unobserved variables following a Gumbel distribution, with location parameters α_i and for simplicity equal scale parameters $\beta > 0$. So that,

$$W_i \quad \sim \quad G(\alpha_i, beta), \qquad \beta > 0$$

then the pdf of W_i, according to [1] is

$$P(W_i \leq w_i) \quad = \quad \exp(-(w_i - \alpha_i)) \cdot \exp\left(-\exp\left(-\frac{w_i - \alpha_i}{\beta}\right)\right).$$

Using this formulation of the latent item variables, the preference probability can be modeled as

$$\begin{aligned}
p_{ij} &= P(Y_{ij} = 1) \\
&= P(W_i \geq W_j) \\
&= P(W_j - W_i \leq 0) \\
&= P\left(\frac{W_j - W_i - (\alpha_j - \alpha_i)}{\beta} \leq \frac{0 - (\alpha_j - \alpha_i)}{\beta}\right) \\
&= \mathrm{LOG}((\alpha_i - \alpha_j)/\beta) \\
&= \frac{1}{1 + \exp(-(\alpha_j - \alpha_i)/\beta)}, \qquad (4.4)
\end{aligned}$$

where the second last line, comes from the fact that the difference of two Gumbel distributed random variables with equal scale parameters, is *logistic* distributed with mean equal to the difference of the Gumbel means and scale parameter equal to the Gumbel scale parameter, see [15] for details.

As β is just a scaling factor we can, without loss of model complexity define a new item parameter θ_i as

$$\theta_i = \frac{\alpha_i}{\beta}, \qquad \forall i = 1, \ldots, t.$$

Using the new item parameter in (4.4)

$$\begin{aligned}
p_{ij} &= \mathrm{LOG}(\theta_j - \theta_i) \quad \Leftrightarrow \\
\log \frac{p_{ij}}{1 - p_{ij}} &= \theta_j - \theta_i \qquad \Leftrightarrow \\
\mathrm{logit}(p_{ij}) &= \theta_j - \theta_i. \qquad (4.5)
\end{aligned}$$

Making another transformation, $\pi_i = \exp(\theta_i) \Leftrightarrow \theta_i = \log(\pi_i)$, the Bradley-Terry formulation in (4.4) is recognized as

$$p_{ij} \quad = \quad \frac{\pi_i}{\pi_i + \pi_j}, \tag{4.6}$$

This result is consistent with the theory for the Rash model for dichotomous data. The Rash model for dichotomous data models the probability of item i being preferred to item j, as a function of both a person parameter of the persons ability to answer correct and of an item parameter. By conditioning on the person to answer correct, the model states that the logarithm of the odds (logit) of $P(Y_{ij} = 1)$ can be modeled as the difference in the item parameters, as seen in (4.5).

4.4 Inference

Given a set of data from a sequence of paired comparisons, how would it then be possible to draw inference using either the TM or the BT model?

Recalling from (4.3) and (4.4) that both models formulate p_{ij} as

$$p_{ij} = \mathrm{g}^{-1}(\theta_j - \theta_i),$$

where the function g in the TM approach is the inverse probit, and in the BT approach is the logit function.

Here it should be stressed, that an assumption about no *order effect*, which means $p_{ij} = 1 - p_{ji}$, has been made. Thereby only observations y_{ij} where $i < j$ are modeled, reducing the degrees of freedom in the model. This gives, in a set of t items, $\binom{t}{2} = \frac{(t-1)t}{2}$ different stochastic variables.

The observation vector y_k for each consumer $k = 1, \ldots, n$, is defined as an observation from the multiple stochastic variable $\mathbf{Y} = (Y_{12}, Y_{13}, \ldots, Y_{t-1,t})$, with one element for each type of paired comparison made by consumer number k.

Assuming that all the paired comparisons $(i < j)$ made by a consumer are independent, the probability of observing y_k can be formulated as the product over all the paired comparisons made, which according to (4.1) each follow a binomial distribution with parameters $(p_{ij}, 1)$,

$$\begin{aligned}
P(Y = y_k) &= \prod_{i=1}^{t-1} \prod_{j=i+1}^{t} P(Y_{ij} = y_{ijk}) \\
&= \prod_{i=1}^{t-1} \prod_{j=i+1}^{t} \binom{1}{y_{ij}} p_{ij}^{y_{ij}} (1 - p_{ij})^{1-y_{ijk}} \\
&= \prod_{i=1}^{t-1} \prod_{j=i+1}^{t} \binom{1}{y_{ijk}} p_{ij}^{y_{ijk}} (1 - p_{ij}) (\frac{1}{1-p_{ij}})^{y_{ijk}} \\
&= \prod_{i=1}^{t-1} \prod_{j=i+1}^{t} \binom{1}{y_{ijk}} \left(\frac{p_{ij}}{1-p_{ij}}\right)^{y_{ijk}} (1 - p_{ij}) \\
&= \prod_{i=1}^{t-1} \prod_{j=i+1}^{t} \binom{1}{y_{ijk}} \left(\frac{p_{ij}}{p_{ji}}\right)^{y_{ijk}} (p_{ji}),
\end{aligned}$$

for all consumers $k = 1, \ldots, n$.

Therefore the total probability of the observed data is

$$\begin{aligned}
P(\mathbf{Y} = \mathbf{y}) &= \prod_{k=1}^{n} P(Y_k = y_k) \\
&= \prod_{k=1}^{n} \prod_{i=1}^{t-1} \prod_{j=i+1}^{t} \binom{1}{y_{ijk}} \left(\frac{p_{ij}}{p_{ji}}\right)^{y_{ijk}} (p_{ji}).
\end{aligned}$$

Maximum Likelihood

The likelihood function is

$$L(\theta; \mathbf{y}) = \prod_{k=1}^{n} \prod_{i=1}^{t-1} \prod_{j=i+1}^{t} \binom{1}{y_{ijk}} \left(\frac{p_{ij}}{p_{ji}}\right)^{y_{ijk}} (p_{ji}), \tag{4.7}$$

and thereby the log-likelihood

$$\begin{aligned}
\ell(\theta; \mathbf{y}) &= \sum_{k=1}^{n} \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} \log \binom{1}{y_{ijk}} + y_{ijk}(\log(p_{ij}) - \log(p_{ji})) + \log(p_{ji}) \\
&= a + \sum_{k=1}^{n} \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} \log(\frac{p_{ij}}{p_{ji}}) + n \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} \log(p_{ji}).
\end{aligned}$$

where $a = \sum_{k=1}^{n} \sum_{i=1}^{t-1} \sum_{j=i+1}^{t} \log \binom{1}{y_{ijk}}$ is a constant with respect to p_{ij} and can therefore be neglected in the maximization.

Using the different distributions of Y_{ij} for the TM model and the BT model, gives different log-likelihood functions ℓ_{TM} and ℓ_{BT}, given as

$$\begin{aligned}\ell_{TM}(\theta;\mathbf{y}) &= a + \sum_{k=1}^{n}\sum_{i=1}^{t-1}\sum_{j=i+1}^{t} y_{ijk} \log\left(\frac{\Phi(d_{ij})}{\Phi(-d_{ij})}\right) + \\ &\qquad n\sum_{i=1}^{t-1}\sum_{j=i+1}^{t} \log(\Phi(-d_{ij})), \end{aligned} \tag{4.8}$$

where the distance $d_{ij} = \theta_i - \theta_j$, and

$$\ell_{BT}(\theta;\mathbf{y}) = a + \sum_{k=i}^{n}\sum_{i=1}^{j-1}\sum_{j=i+1}^{t} y_{ijk} \frac{\pi_i}{\pi_j} + n\sum_{i=1}^{j-1}\sum_{j=i+1}^{t} \frac{\pi_j}{\pi_i + \pi_j}, \tag{4.9}$$

where π is defined like in (4.6), so that $\pi_i = \log(\theta_i)$ for all items $i = 1, \ldots, t$.

Taking the non-linearity of the problem into account, one way to optimize the log-likelihood functions could be through the iterative method of Newton-Raphson, or maybe using a direct search method, that does not need knowledge about the derivatives. Such a directive search method is implemented in the MatLab function `fminsearch`.

Example

As an illustration consider the data from example 1 ind [5], in which 15 persons examine all possible pairings of four different taste samples. The data is given in Table 4.1.

index (i,j)	y_{ij}
$(1,2)$	3
$(1,3)$	2
$(1,4)$	2
$(2,3)$	11
$(2,4)$	3
$(3,4)$	5

Table 4.1: Paired Comparison data from example 1 in [5]. 15 persons have examined all possible pairings of four different taste samples.

The log-likelihood function for the Bradley-Terry model has been implemented in MatLab. The code is found in the file `loglikeBT.m` in appendix C.

The estimated item parameters are found to be

$$\boldsymbol{\theta}_{TM} \quad = \quad (-2.3571, -0.7440, -1.0561, 0),$$

which equals the estimates in [5].

An other way of estimating the item parameters could be to recognize the model for the problem as a GLM, and then use an IRLS method, e.g. the one implemented in the software R, in the function called `glm`.

GLM approach

As the components of the multiple stochastic variable $\mathbf{Y} = (Y_{12}, Y_{13}, \ldots, Y_{t-1,t})$, are independent and binomial distributed, $\mathbf{Y}$ can be modeled as a generalized linear model with binomial distribution, according to section 2.

According to (4.7) both the TM model and the BT model has the following model of the mean

$$\begin{aligned} g(p) &= \mathbf{X}\boldsymbol{\theta} \Leftrightarrow \\ g(\frac{\mu}{n}) &= \mathbf{X}\boldsymbol{\theta}, \end{aligned}$$

where the $\boldsymbol{\theta} = (\theta_i, \ldots, \theta_t)^\intercal$ is the parameter vector and the model matrix $\mathbf{X}$ is given as

$$\mathbf{X}_{ij,k} = \begin{cases} 1 & \text{if } k = i, \\ -1 & \text{if } k = j, \\ 0 & \text{otherwise,} \end{cases}$$

which is a matrix with one row per paired comparison made. That is $\binom{t}{2} = \frac{(t-1)(t)}{2}$ rows. Each row has a 1-entry in the i'th column and a -1-entry in the j'th column.

The link function $g(p)$ is recognized as the (scaled) probit link (2.6) for TM model, and the (scaled) logit link (2.5)for the BT model.

Now both models are formulated as Generalized Linear Models and an IRLS method, can be used to estimate the parameters.

Code for estimating the item parameters for data from Table 4.1, using a GLM framework with binomial distribution and logit as well as probit link function is presented in `binprobitlogitCFeks1.R` in appendix D.

The estimated item parameters are found to be

$$\begin{aligned}\boldsymbol{\theta}_{\text{logit}} &= (-2.3571, -0.7441, -1.0561, 0) \quad \text{and} \\ \boldsymbol{\theta}_{\text{probit}} &= (-1.3874, -0.4421, -0.6192, 0),\end{aligned}$$

which equals the estimates in [5].

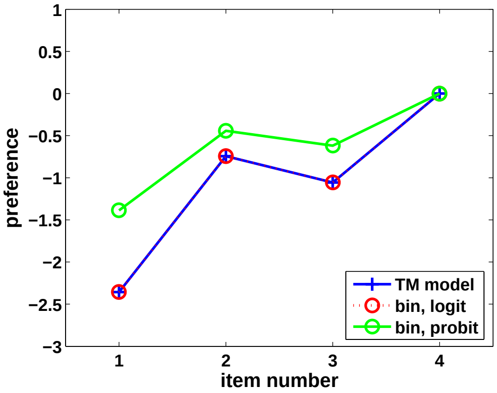


Figure 4.1: Estimated values of θ, with GLM probit, GLM logit and ML-estimate.

In Figure 4.1 the estimated item parameters are plotted, both the maximum likelihood estimate, and the GLM estimates with probit and logit link functions. Notice that the estimate with logit link and the ML-estimate are equal, as assumed.

Chapter 5

Ranking Models

In section 4 models for estimating item parameters from paired comparison data was described. The next two sections will be concerning models for estimating the item parameters from ranking data.

First ranking data will be described followed by a brief overview of the diversity of ranking models, and then in section 6, ranking models based on paired comparisons will be treated.

5.1 Ranking Data

Assuming that the item parameters of t items are to be estimated. The preferences of a panel of n consumers are used as observations. The consumers are each asked to order the t items, starting with the one they like the most ending with the one they like the least.

The consumers are of cause assumed to have equal preference scale, so that the observations can be treated as independent outcomes of the same distribution.

Notice the difference between the concepts *ordering* and *ranking*. A *ranking* of

t items, is defined as

$$r_u = (r_{1u}, r_{2u}, \ldots, r_{tu}) \qquad \forall u = 1, \ldots, t!,$$

where r_{iu} is the rank value chosen for item i in ranking number u, while the *ordering* of the items is defined as

$$h_u = \langle h_{1u}, h_{2u}, \ldots, h_{tu} \rangle \qquad \forall u = 1, \ldots, t!,$$

where h_{iu} is the item number of the item with rankvalue i in ranking number u.

The relation between a ranking r_u and ordering h_u is unique, and therefor the data could be observed in either ways.

All possible rankings of t items can be described by all possible permutations of the indexes of the items. That is $t!$ different rankings of t items.

To derive a mathematical model of how to estimate item parameters from a set of ranking observations, the stochastic variable Y_{uk} of the ranking observations is defined;

$$Y_{uk} = \begin{cases} 1 & \text{if consumer } k \text{ rank the items according to ranking } r_u \\ 0 & \text{otherwise,} \end{cases}$$

for all $u = 1, \ldots, t!$ and $k = 1, \ldots, n$.

Written in another way

$$Y_{uk} \quad \sim \quad \text{bin}(p_u, 1) \qquad \forall\ u = 1, \ldots, t!,$$

where $p_u = P(Y_{uk} = 1)$.

Since a consumer must prefer one and only one ranking to all the others, the probabilities p_u , for $u = 1, \ldots, t!$ must sum to one,

$$\sum_{u=1}^{t!} P(R_u) = 1,$$

where R_u is used as a notation for the event that a consumer rank the items according to ranking r_u, $\{R_u\} = \{Y_{uk} = 1\}$.

So far all ranking models must agree, but similar to hte PC models in section 4, different approaches to describe the probability p_u has been made through the years.

5.2 The Diversity of Ranking Models

Every author in this field has his or her own way of categorizing ranking models. Therefore the reader is asked to keep in mind, that under the right assumptions the categories presented here might overlap and/or define new categories.

The author find the categories used here very instructive in providing an overview of the models, emphasizing the differences, but as least as important the similarities.

Proportional Odds Models

These models have several names; Order Statistics Models, Proportional Odds Models or Cummulative Logit Models.

Examples of those models are the Thurstone-Mosteller-Dianiels (TMD) model and the Luce model. Both these models are generalizations of the two-paired comparison models Thurstone-Mosteller and Bradley-Terry.

Like in the paired comparison models the perception of an item is modeled as a latent stochastic variable, modeled according to a scaling model.

A simple introduction could be to let q_i be the probability that item i is preferred to all other items.

The probability of item i to be ranked number 1 can then be derived as

$$P(r_{iu} = 1) \quad = \quad q_i.$$

Then the probability of item j to be ranked number 2, given that item i is ranked number 1, must be

$$P(r_{ju} = 2 |\ r_{iu} = 1) \quad = \quad \frac{q_j}{1 - q_i}.$$

Generalizing this idea, the probability of ranking r_u can be derived as a product of the independent conditional probabilities

$$P(R_u) \quad = \quad \prod_{j=1}^{t} \frac{q_{h_{ju}}}{\sum_{i=j}^{t} q_{h_{iu}}},$$

or written by rankings

$$P(R_u) \quad = \quad \prod_{j=1}^{t} q_j \left(\sum_{i=1, r_{iu} > r_{ju}}^{t} q_i \right)^{-1} .$$

For different introductions see [16], [21] and [18], but for a more thorough review see [1, ch. 7.2.2].

The multistage ranking models

The multistage ranking models was first presented in [8]. These models break the ranking process down into a sequence of $t-1$ stages, like the Luce and TMD models just described.

The difference is that these models are parameterized by the probability of the different degree of "accuracy" at each stage. It is assumed that the accuracy of the choice at any stage is independent of the accuracy at other stages.

Accuracy is assessed with respect to a central ranking. This means that once a central ranking has been chosen, the choice probability at a given stage depends only on the stage rather than on the objects remaining at that stage, contrary to the proportional odds models.

This will, according to [8] avoid analytical difficulties, when the number of items increase to more that "a few".

Distance based ranking models

In these kind of ranking models each ranking $r_u = (r_{1u}, \ldots, r_{tu})$ is looked upon as a point in some t dimensional space, with some distance function attached to the space. The probability of a ranking is found as a function of the distance between the specific ranking and some modal ranking. A ranking has decreasing probability with increasing distance from the modal ranking.

Examples of such models are the Kendall's Distance model an Mallow's ϕ-model. The latter is presented in [7] The "distance" is defined by a metric on permutations, $d(r_u, r_v)$ which is the minimum number of transpositions required to bring r_u to r_v. For an instructive description see [9].

Some generalizations to these models was made by Fligner and Verducci and presented in [7].

Models based on paired comparisons

Yet another kind of models are the models based on paired comparison models. These models are not generalizations of the paired comparison models, in the same sense as the proportional odds models TMD and the Luce-model. These models converts the ranked data into paired comparison data, and then work on it, as if it was paired comparison data.

Example: *A consumer has ordered four items in this preference order: $h_u = \langle 3, 1, 2, 4 \rangle$ meaning that he prefers item 3 the most and 4 the least. This means that the consumer contribute to the data by the ranking $r_u = (2, 3, 1, 4)$. The paired comparison probabilities that are present with this ranking would be p_{12}, p_{14}, p_{24}, p_{31}, p_{32} and p_{34}. One way to describe those probabilities, could be all p_{ij} where $i, j = 1, \ldots, t$ and $r_{iu} < r_{ju}$.*

These models are the main focus of this thesis and different examples of such are presented in the next section 6.

Chapter 6

PC based Ranking Models

In the last section some different approaches to model ranking data were briefly presented. This section will go more thorough into an approach based on the paired comparison (PC) models described in section 4. Three models will be presented.

The first (MBT) mentioned in [5] will describe the data well, but will not be a GLM. The second model BTL, is presented in [4] and identified as a GLM, but it is showed to have some simplifications that does not match the structure of the data. At last a GLM model is derived having the same reasonable parameter estimates as the MBT model.

Even though the models presented here could use either kind of paired comparison models as a reference, this presentation will be limited to use the Bradley-Terry model. In section 4 it was shown that the difference between the BT model and the Thurstone model is quite small both in theory and practice. It will be easy for the reader to replace the Bradley-Terry form of p_{ij}, by any other if wanted, but the BT form makes the equations more readable and easy to work with and is therefore chosen here.

From (4.6) the Bradley-Terry probability of a consumer preferring item i to item

j is

$$p_{ij} \quad = \quad \frac{\pi_i}{\pi_i + \pi_j}.$$

The general idea behind the models is to think of each ranking as a combination of some paired comparisons. This way of thinking of the origin of a ranking might make sense to a psychometrician as a natural way of ranking items. When a person is to rank say 10 items, it is very unlikely that he is able to compare all items at one time. Intuitively one would assume that he, in his head, compare the items two at a time, and then build up his ranking from those comparisons.

An advantage of thinking of the rankings as originating from paired comparisons is that there is a straight forward way to handle incomplete data. Even though a full ranking has not been fully completed by each consumer, still some components of the ranking can be identified as paired comparisons and the information will not be lost. Incomplete data will not be an issue in this thesis.

6.1 Mallows-Bradley-Terry model

Recalling from section 5 the structure of a ranking of t items

$$r_u = (r_{1u}, r_{2u}, \dots, r_{tu}),$$

and the stochastic Bernoulli variable Y_{uk} describing whether consumer $k = 1, \dots, n$ rank the items according to ranking r_u for $u = 1, \dots, t!$ or not.

$$Y_{uk} \quad \sim \quad \text{bin}(p_u, 1),$$

where the probability $p_u = P(R_u)$, according to 5.1.

A way to define all the different paired comparisons being consistent with some ranking r_u, is to chose all the pairs (i, j) where the rank value of item i is smaller than the rank value of item j, that is $r_{iu} \leq r_{ju}$. Assuming independence between the pairs, this leads to a description of the probability that the ranking r_u, is observed in a one-trial experiment

$$P(R_u) \quad = \quad c_0(\boldsymbol{\theta}) \prod_{(i,j), r_{iu} < r_{ju}}^{t} p_{ij}, \quad u = 1, \dots, t!$$

where $c_0(\boldsymbol{\theta})$ is a normalization constant, assuring that $\sum_{u=1}^{t!} P(R_u) = 1$.

Now following the Bradley-Terry theory defining, $p_{ij} = \frac{\pi_i}{\pi_i+\pi_j}$, one can write

$$\begin{aligned}
P(R_u) &= c_0(\boldsymbol{\theta}) \prod_{(i,j),r_{iu}<r_{ju}}^{t} \frac{\pi_i}{\pi_i+\pi_j} \\
&= c_0(\boldsymbol{\theta}) \frac{\prod_{i=1}^{t} \pi_i^{t-r_{iu}}}{\prod_{i=1}^{t-1}\prod_{j=i+1}^{t}(\pi_i+\pi_j)} \\
&= c_0(\boldsymbol{\theta}) \frac{\prod_{i=1}^{t} \pi_i^{t-r_{iu}}}{\pi_0} \\
&= c(\boldsymbol{\theta}) \prod_{i=1}^{t} \pi_i^{t-r_{iu}}, \qquad (6.1)
\end{aligned}$$

where the two last lines comes from defining the constant $\pi_0 = \prod_{i=1}^{t-1}\prod_{j=i+1}^{t}(\pi_i+\pi_j)$, and a new normalization constant $c(\boldsymbol{\theta}) = c_0(\boldsymbol{\theta})/\pi_0$.

The Normalization Constant

To ensure that a consumer prefer one and only one ranking to all the other rankings, the sum of the ranking probabilities over all rankings has to sum to 1, $\sum_{u=1}^{t!} P(R_u) = 1$. This is made true by the normalization constant $c(\boldsymbol{\theta})$, depending on the parameters.

Using (6.1) to describe $P(R_u)$ the constant can be derived from

$$\begin{aligned}
1 &= \sum_{u=1}^{t!} P(R_u) \\
&= \sum_{u=1}^{t!} c(\boldsymbol{\theta}) \prod_{i=1}^{t} \pi_i^{t-r_{iu}} \\
&= c(\boldsymbol{\theta}) \sum_{u=1}^{t!}\prod_{i=1}^{t} \pi_i^{t-r_{iu}} \qquad \Leftrightarrow \\
c(\boldsymbol{\theta}) &= \left(\sum_{u=1}^{t!}\prod_{i=1}^{t} \pi_i^{t-r_{iu}}\right)^{-1}.
\end{aligned}$$

Distribution of Y

Assuming that the number of consumers n is constant, the stochastic variable Y_u of the sum of the Bernoulli variables Y_{uk} over consumers, must be defined with distribution

$$Y_u = \sum_{k=1}^{n} Y_{uk} \quad \sim \quad \text{bin}(p_u, n), \qquad u = 1, \ldots, t!, \tag{6.2}$$

which can also be thought of as a stochastic variable for the number of times the ranking r_u will occur in the data set.

Since $\sum_{u=1}^{t!} p_u = 1$, and $Y_u \sim \text{bin}(p_u, n)$, the multiple stochastic variable $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_{t!})$ must be polynomial distributed according to [10], with parameters given as

$$\mathbf{Y} \quad \sim \quad \text{poly}((p_1, p_2, \ldots, p_{t!}), n),$$

with probability density function

$$\begin{aligned} P(\mathbf{Y} = \mathbf{y}) &= P(Y_1 = y_1 \wedge Y_2 = y_2 \wedge \cdots \wedge Y_{t!} = y_{t!}) \\ &= \binom{n}{y_1 \; y_2 \cdots y_{t!}} \prod_{u=1}^{t!} P(Y_u = y_u) \\ &= \binom{n}{y_1 \; y_2 \cdots y_{t!}} \prod_{u=1}^{t!} P(R_u)^{y_u}. \end{aligned}$$

6.1.1 Parameter estimation

Even though the components of $\mathbf{Y}$ can be written as being distributed according to the exponential family, the model does not fall into the frames of GLM. The reason is that in the polynomial distribution, the components are not independent, and therefor the GLM acquirements are not fulfilled. Inference can therefor not be done by use of the IRLS method described in 2 , but must be done by iteratively maximizing the likelihood.

Another approach could be to use of a multiple version of IRLS, since the polynomial distribution can be thought of as a Multivariate Generalized Linear Model. Those models will not be handled in this thesis, but an introduction can be found in [1].

The likelihood function

$$\begin{aligned} L_{\mathrm{MBT}}(\boldsymbol{\theta};\mathbf{y}) &= \binom{n}{y_1\ y_2 \cdots y_{t!}} \prod_{u=1}^{t!} P(R_u)^{y_u} \\ &= \binom{n}{y_1\ y_2 \cdots y_{t!}} \prod_{u=1}^{t!} \left(c(\boldsymbol{\theta}) \prod_{i=1}^{t} \pi_i^{t-r_{iu}} \right)^{y_u} . \end{aligned}$$

The log-likelihood function

$$\begin{aligned} \ell_{\mathrm{MBT}}(\boldsymbol{\theta};\mathbf{y}) &= \log\left(\binom{n}{y_1\ y_2 \cdots y_{t!}} \prod_{u=1}^{t!} \left(c(\theta) \prod_{i=1}^{t} \pi_i^{t-r_{iu}} \right)^{y_u} \right) \\ &= b_0 + \sum_{u=1}^{t!} y_u \log\left(c(\boldsymbol{\theta}) \prod_{i=1}^{t} \pi_i^{t-r_{iu}} \right) \\ &= b_0 + \sum_{u=1}^{t!} y_u \left(\log(c(\boldsymbol{\theta})) + \sum_{i=1}^{t} (t-r_{iu})\theta_i \right) \\ &= b_0 + n\log(c(\boldsymbol{\theta})) + \sum_{u=1}^{t!} y_u \left(\sum_{i=1}^{t} (t-r_{iu})\theta_i \right) \qquad (6.3) \end{aligned}$$

where $b_0 = \log \binom{n}{y_1\ y_2 \cdots y_{t!}}$, and $\pi_i = \exp\theta_i$.

As an illustration consider the data from example 3 in [5], in which 32 consumers have ranked four salad dressings. The data is given in Table 6.1.

u	$(r_i$	r_2	r_3	$r_4)$	y_u
1	1	2	3	4	2
2	1	2	4	3	0
3	1	3	2	4	0
4	1	3	4	2	0
5	1	4	2	3	0
6	1	4	3	2	0
7	2	1	3	4	1
8	2	1	4	3	2
9	2	3	1	4	1
10	2	3	4	1	0
11	2	4	1	3	0
12	2	4	3	1	0
13	3	1	2	4	2
14	3	1	4	2	1
15	3	2	1	4	0
16	3	2	4	1	0
17	3	4	1	2	0
18	3	4	2	1	1
19	4	1	2	3	11
20	4	1	3	2	6
21	4	2	1	3	3
22	4	2	3	1	0
23	4	3	1	2	1
24	4	3	2	1	1

Table 6.1: Ranking data from example 3 in [5]. 32 consumers have ranked 4 salad dressings according to their preference.

The loglikelihood function of the MBT model has been implemented in MatLab, and using the direct search algorithm `fminsearch`, the following estimates of the item parameters has been found.

$$\boldsymbol{\theta}_{MBT} \quad = \quad (-0.5549, 1.1824, 0.3776, 0).$$

Notice that θ_1 has the lowest value, followed by θ_4, θ_3 and θ_2. This gives an estimated order $\tilde{h} = \langle 1, 4, 3, 2 \rangle$, equivalent to the ranking $\tilde{r} = (4, 1, 2, 3)$, which is the ranking most consumers have chosen in the data in Table 6.1.

6.2 Bradley-Terry-Luce model

As an alternative to the MBT model, [4] presents a ranking model inspired by the Bradley-Terry model for paired comparison. The advantage of this model is that it falls into the frames of GLM. In the following section the model will be described in a more mathematical consistent way, than what is found in [4], highlighting the assumptions and simplifications made.

Consider the stochastic Bernoulli distributed stochastic variable Y_{ijk} from section 4,

$$Y_{ijk} = \begin{cases} 1 & \text{consumer } k \text{ prefer item } i \text{ to item } j \\ 0 & \text{otherwise,} \end{cases}$$

where $P(Y_{ijk} = 1) = p_{ij}$.

Defining $t(t-1)/2$ new stochastic variables by summing over the n consumers or observations one gets the variables $Y_{ij} = \sum_{k=1}^{n} Y_{ijk}$ for $i < j$ and $i, j = 1, \ldots, t$. The distributions of these variables are binomial with parameters like

$$Y_{ij} \sim bin(p_{ij}, n),$$

with mean $E(Y_{ij}) = np_{ij}$. Y_{ij} describes the number of times item i is preferred to item j in the ranking data set.

At this point Courcoux and Semenou makes an assumption about the stochastic variables Y_{ij} being independent. Even though the dependency between them is more complex than for the variables Y_u, in the MBT case, it is not correct to assume independence between them.

In other words, it still holds that the sum of the rank-probabilities must sum to one, and since the probabilities p_{ij} are factors of the rank probabilities, according to (6.1), there is a relation between them, and they should not be assumed independent. Just as well as the multiple stochastic variable in the MBT model $\mathbf{Y} = (Y_1, \ldots, Y_{t!})$ must be polynomial distributed and not multiple-binomial.

To assume independence between the Y_{ij}'s, is the same as allowing for cycles in the data, as might be possible for real paired comparison data, but when the data originates from ranked data, the consumers have already made sure that no cycles are possible.

A consumer in a paired comparison experiment might prefer item 1 to item 2, item 2 to item 3 and item 3 to item 1. This is possible as the consumers preferences might be influenced by order or carry-over effects. Such effects are not possible ind ranking data. This is because in ranking tests the consumer is forced to make a full ranking ranking of the items, not only pair by pair.

A consequence is that the BTL model expect a larger variance in the observations, than what will actually be found in data.

This drawback of the simplification is not mentioned by [4], but it is clear why the simplification is made. The stochastic variable $\mathbf{Y}$ has binomial independent components, and can therefore fit into the GLM framework.

The probability density function of the stochastic variable $\mathbf{Y} = (Y_{12}, Y_{13}, \ldots, Y_{(t-1),t})$ will take the form

$$P(\mathbf{Y} = \mathbf{y}) \quad = \quad \prod_{i=1}^{n-1} \prod_{j=i+1}^{n} P(Y_{ij} = y_{ij}),$$

The BTL model can then be described as a transformation of the ranked data into paired comparison data with loss of the no-cycling condition in the ranked data, and then using the BT model on the transformed data.

6.2.1 Parameter Estimation

The likelihood function for the model has a lot of similarity to the the one for the BT model given in (4.7),

$$L(\boldsymbol{\theta}; \mathbf{y}) \quad = \quad \prod_{k=1}^{n} \prod_{i=1}^{t-1} \prod_{j=i+1}^{t} \binom{n}{y_{ijk}} \left(\frac{p_{ij}}{p_{ji}}\right)^{y_{ijk}} (p_{ij}),$$

where $p_{ij} = \pi_i/(\pi_i + \pi_j)$ and $\theta_i = \log(\pi_i)$ for all $i, j = 1, \ldots, t$.

The log-likelihood function then follows as,

$$\begin{aligned} \ell_{BTL}(\boldsymbol{\theta}; \mathbf{y}) \quad &= \quad a_0 + \sum_{k=i}^{n} \sum_{i=1}^{j-1} \sum_{j=i+1}^{t} y_{ijk} \frac{\pi_i}{\pi_j} + n \sum_{i=1}^{j-1} \sum_{j=i+1}^{t} \frac{\pi_j}{\pi_i + \pi_j} \\ &\propto \quad \sum_{k=i}^{n} \sum_{i=1}^{j-1} \sum_{j=i+1}^{t} y_{ijk} \frac{\pi_i}{\pi_j} + n \sum_{i=1}^{j-1} \sum_{j=i+1}^{t} \frac{\pi_j}{\pi_i + \pi_j}, \end{aligned}$$

where the constant $a_0 = \sum_{k=i}^{n} \sum_{i=1}^{j-1} \sum_{j=i+1}^{t} \log \binom{n}{y_{ijk}}$., and $\theta_i = log(\pi_i)$, for all $i = 1, \ldots, t$.

The estimates of the parameters θ can be found by maximization of the log-likelihood using e.g. the `fminsearch` function in MatLab.

This procedure has been tested on the data from [5], given in Table 6.1. Code for an implemented in MatLab of the loglikelihood function is given in the file `loglikeBTL.m` in appendix C, where also the file with the estimation procedure `testlikelihood.m` is found.

The estimated item parameters are found to be

$$\boldsymbol{\theta}_{BTL} \quad = \quad (-0.8271, 1.8147, 0.6116, 0).$$

In Figure 6.1 the estimates from maximizing ℓ_{MBT} and ℓ_{BTL} are compaired. An analysis of the practical error using BTL in stead of MBT would be of high interest in further work.

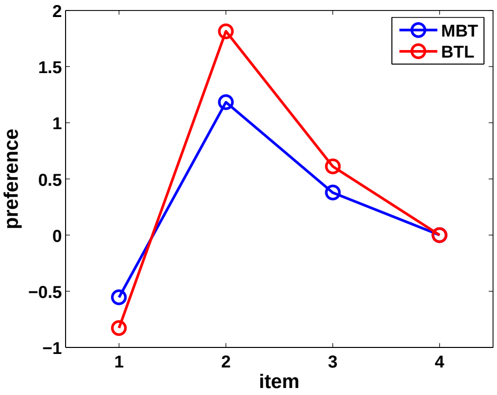


Figure 6.1: Estimated values of $\boldsymbol{\theta}$, found using the MBT model and the BTL model.

GLM approach

Since the BTL model equals the BT model, the same GLM holds for both models, each component Y_{ij} of the multible stochastic variable $\mathbf{Y}$ is binomial distributed with probability p_{ij} and quantity parameter n, and they are assumed independent according to the mentioned simplification.

This gives a GLM with binomial distribution, (scaled) logit link function, and a linear predictor $\mathbf{X}\boldsymbol{\theta}$ like the one in the paired comparison BT model in (4.10).

Parameter estimation by call to the R function `glm`, has been done. The code is found in `binprobitlogitCFeks3.R` presented in appendix E.

The estimated item parameters are found to be

$$\begin{aligned} \boldsymbol{\theta}_{logit} &= (-0.8271, 1.8147, 0.6116, 0) \quad \text{and} \\ \boldsymbol{\theta}_{probit} &= (-0.4740, 1.0654, 0.3689, 0). \end{aligned}$$

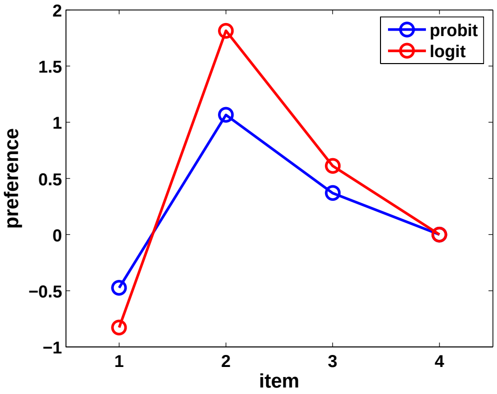


Figure 6.2: Estimated values of $\boldsymbol{\theta}$, found using the GLM framework with logit and probit link.

In Figure 6.2 the estimated item parameters plotted. Notice that the estimates found using the logit link function in the GLM framework are equal to the estimates found by maximizing the loglikelihood function of the BTL model, as would be expected.

6.3 Ranking model with poisson

In [5] the MBT model was presented. In the article it is claimed that the MBT model can easily be written as a GLM, which according to section 6.1 is not possible since the components of a polynomial distributed stochastic variable are not independent.

However in the appendix of [5] some code for a call to some GLM software called *GLIM* was listed. The code indicated that a poisson model was used with a log link function. This showed that it was not the MBT model but another model, which was used within the GLM framework.

This section will describe the mathematical derivation of a model, which is not the MBT model, but which is found to give the same estimates, but contrary to the MBT it holds the ability of being written as a GLM.

Recall from (5.1) the stochastic bernoulli variable

$$Y_{uk} = \begin{cases} 1 & \text{consumer } k \text{ prefer ranking } r_u \\ 0 & \text{otherwise,} \end{cases}$$

where $P(Y_{uk} = 1) = p_u$, and from (6.2) the stochastic variables for the number of times ranking r_u occur in the data was assumed to be binomial distributed with parameters p_u and n, since the number of consumers was assumed to be a known constant. The expected value of Y_u being $E(Y_u) = np_u$.

Distribution of Y

Once again the multiple stochastic variable $\mathbf{Y} = (Y_1, \ldots, Y_t!)$ is defined, but now another approach to model the components Y_u will be taken. Assuming that the number n_u of consumers evaluating rank r_u, is an (unknown) variable, Y_u will follow a Poisson distribution. Though still assumed to have the "correct" binomial mean $E(Y_u) = np_u$. Notice that now, the components of $\mathbf{Y}$ can be assumed independent, even since the sum of the probabilities must still sum to 1.

Therefore now

$$P(Y_u = y_u) \quad = \quad e^{-\lambda_u} \cdot \frac{\lambda_u^{y_u}}{y_u!}$$

with mean $\lambda = nP(R_u)$, inherited by the polynomial MBT model.

Due to the assumed independency of the components, the total pdf for the data can be written as

$$\begin{aligned} P(\mathbf{Y}=\mathbf{y}) &= \prod_{u=1}^{t!} P(Y_u=y_u) \\ &= \prod_{u=1}^{t!} e^{-nP(R_u)} \cdot \frac{(nP(R_u))^{y_u}}{y_u!}. \end{aligned} \tag{6.4}$$

6.3.1 Parameter Estimation

The parameter estimation is once again first done by maximizing the log-likelihood ℓ_{Poiss}. The derivation of the log-likelihood will show that it is proportional to the log-likelihood of the MBT model.

Secondly the new model will be found to fall into the GLM framework, and thereby being solvable by means of an IRLS algorithm.

Maximum Likelihood

The likelihood function is given by the probability density function (6.4)

$$\begin{aligned} L(\boldsymbol{\theta};\mathbf{y}) &= \prod_{u=1}^{t!} P(Y_u=y_u) \\ &= \prod_{u=1}^{t!} e^{-nP(R_u)} \cdot \frac{(nP(R_u))^{y_u}}{y_u!} \\ &= \prod_{u=1}^{t!} e^{-nP(R_u)} \cdot \frac{(nc(\boldsymbol{\theta})\prod_{i=1}^{t}\pi_i^{t-r_{iu}})^{y_u}}{y_u!}, \end{aligned}$$

where $\pi_i = \exp(\theta_i)$.

The Log-likelihood function of $\mathbf{Y}$ is therefore

$$\begin{aligned}
\ell_{\text{Poiss}}(\boldsymbol{\theta};\mathbf{y}) &= \sum_{u=1}^{t!}(-nP(R_u)) + \log(\prod_{u=1}^{t!}\frac{1}{y_u!}) + \\
&\qquad \sum_{u=1}^{t!} y_u \left(\log(nc(\boldsymbol{\theta}) + \sum_{i=1}^{t}(t-r_{iu})\log(\pi_i) \right) \\
&= -n + \log(\prod_{u=1}^{t!}\frac{1}{y_u!}) + n\log(nc(\boldsymbol{\theta})) + \sum_{u=1}^{t!} y_u \left(\sum_{i=1}^{t}(t-r_{iu})\theta_i \right) \\
&= -n + \log\left(\frac{1}{n!} \binom{n}{y_1 \; y_2 \cdots y_{t!}} \right) + \\
&\qquad n\log(n) + n\log(c(\boldsymbol{\theta})) + \sum_{u=1}^{t!} y_u \left(\sum_{i=1}^{t}(t-r_{iu})\theta_i \right) \\
&= -n - \log(n!) + b_0 + n\log(n) + n\log(c(\boldsymbol{\theta})) + \sum_{u=1}^{t!} y_u \left(\sum_{i=1}^{t}(t-r_{iu})\theta_i \right),
\end{aligned}$$

where the constant b_0 is defined like in (6.3), as $b_0 = \log \binom{n}{y_1 \; y_2 \cdots y_{t!}}$.

Comparing the log-likelihood of the MBT model (6.3), with the one found for the new model, it is seen that

$$\begin{aligned}
\ell_{\text{Poiss}}(\boldsymbol{\theta};\mathbf{y}) &= -n - \log(n!) + n\log(n) + \ell_{\text{MBT}}(\boldsymbol{\theta};\mathbf{y}) \\
&\propto \ell_{MBT}(\theta;\mathbf{y}),
\end{aligned}$$

which means that the new model gives exactly the same estimates of the item parameters θ as the MBT model.

This was confirmed by implementing ℓ_{Poiss} in the MatLab function `loglikepoiss.m` (see appendix C and maximize it using the build in MatLab function `fminsearch`. Code for this estimation is found in `testlikelihood.m`, also presented in appendix C.

IRLS

The model derived above has the advantage that the item parameter estimates are "correct" in the sense of being equal to the ones estimated using MBT. This section will show how the new model can be recognized as a GLM, and therefor through GLM software, an IRLS algorithm can be used to estimate the parameters.

According to section 2, the components of $\mathbf{Y}$ must be independent and distributed according to the exponential family of distributions, if the model should be characterizes as a GLM. The independence of the components Y_u for $u = 1, \dots, t!$ has just been explained, and it was shown in section 2 that the Poisson distribution is a member of the exponential family of distributions, with the functions defined as $a(\phi) = 1$, $b(\eta_u) = \exp(\eta_u)$ and $c(y_u, \phi) = -\log(y_u!)$.

The choice of mean equal to the MBT mean gives

$$\begin{aligned} E(Y_u) &= \lambda_u \\ &= nP(R_u) \\ &= nc(\boldsymbol{\theta}) \prod_{i=1}^{t} \pi_i^{t-r_{iu}} \\ &= \exp\left(\log(nc(\boldsymbol{\theta})) + \log(\prod_{i=1}^{t} \pi_i^{t-r_{iu}}) \right) \\ &= \exp\left(\log(nc(\boldsymbol{\theta})) + \sum_{i=1}^{t} (t - r_{iu})\theta_i \right). \end{aligned}$$

This makes it possible to write the logarithm of the expected value of Y_u for all $u = 1, \dots, t!$

$$\begin{aligned} \log(\lambda_u) &= \log(nc(\boldsymbol{\theta})) + \sum_{i=1}^{t} (t - r_{iu})\theta_i, \\ &= \vartheta + \sum_{i=1}^{t} (t - r_{iu})\theta_i, \end{aligned}$$

defining a new normalization parameter $\vartheta = \log(nc(\boldsymbol{\theta}))$.

Thereby the linear system of equations look like

$$\begin{aligned} \begin{pmatrix} \log(\lambda_1) \\ \log(\lambda_2) \\ \vdots \\ \log(\lambda_{t!}) \end{pmatrix} &= \begin{pmatrix} t - r_{11} & t - r_{21} & \dots & t - r_{t1} & 1 \\ t - r_{12} & t - r_{22} & \cdots & t - r_{t2} & 1 \\ \vdots & & \ddots & \vdots & \vdots \\ t - r_{1t!} & t - r_{2t!} & \dots & t - r_{tt!} & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_t \\ \vartheta \end{pmatrix}, \\ \log(\lambda) &= [\mathbf{X}|\mathbf{1}] \begin{bmatrix} \boldsymbol{\theta} \\ \vartheta \end{bmatrix} \end{aligned}$$

where the last equation defines the $n \times t$ matrix $\mathbf{X}$ with elements

$$\{\mathbf{X}\}_{ui} = t - r_{iu}.$$

An estimate of the item parameters θ using an IRLS approach can be executed by calling the R function `glm`.

Call to glm - Salad data

```
     >y=c(2,0,0,0,0,0,1,2,1,0,0,0,2,1,0,0,0,1,11,6,3,0,1,1)
     >x1=4-c(1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4)
     >x2=4-c(2,2,3,3,4,4,1,1,3,3,4,4,1,1,2,2,4,4,1,1,2,2,3,3)
5    >x3=4-c(3,4,2,4,2,3,3,4,1,4,1,3,2,4,1,4,1,2,2,3,1,3,1,2)
     >x4=4-c(4,3,4,2,3,2,4,3,4,1,3,1,4,2,4,1,2,1,3,2,3,1,2,1)

     # model
     >model <- glm(y ~ x1+x2+x3+x4, family=poisson(link=log))
10   >summary(model)

     #OUTPUT:
     #=========

15   Call:
     glm(formula = y ~ x1 + x2 + x3 + x4, family = poisson(link = log))

     Deviance Residuals:
         Min       1Q   Median       3Q      Max
20   -1.9410  -0.6043  -0.3296   0.3676   2.0695

     Coefficients: (1 not defined because of singularities)
                 Estimate Std. Error z value Pr(>|z|)
     (Intercept)  -2.3097     1.0390  -2.223   0.0262 *
25   x1           -0.5548     0.2491  -2.228   0.0259 *
     x2            1.1825     0.2866   4.127 3.68e-05 ***
     x3            0.3776     0.2228   1.694   0.0902 .
     x4                NA         NA      NA       NA
     ---
30   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     (Dispersion parameter for poisson family taken to be 1)

         Null deviance: 70.753  on 23  degrees of freedom
35   Residual deviance: 22.249  on 20  degrees of freedom
     AIC: 60.99

     Number of Fisher Scoring iterations: 5
```

Notice (line 8) that the call is made with design matrix $\mathbf{X}$ and not $[\mathbf{X}|\mathbf{1}]$, because the normalization parameter ϑ is taking care of within the `glm` function.

The estimates of $\boldsymbol{\theta}$ can be read from the output to be equal to the estimates from inference of the MBT model, as it was expected.

An other thing to notice about the design matrix is the size of it as a function of the item number. The design matrix $\mathbf{X}$ has $t!$ rows. With eg. 10 items to be compared, $\mathbf{X}$ is a non-sparse matrix of size $3,628,800 \times 10$. In a test executed, it was not possible to get `glm` to work on such a huge matrix, even with a memory limit at 4 GB.

The lefthand side of the system, though is very sparse. The number of non-zero elements are at most equal to the number of consumers n. An interesting thing to look into at this point would be how to enhance the sparse structure of the lefthand side to reduce the number of equations. Right now it is not possible since every row holds a contribution to the normalization constant. But reducing the size of the system is definitely a tempting advancement of the model to work on.

If the `glm` function was used to fit the model of the data simplified by removing the zeros from the lefthand sid eog the system, the estimates change.

Call to glm - Modified Salad Data

```
>y=c(2,1,2,1,2,1,1,11,6,3,1,1)
>x1=4-c(1,2,2,2,3,3,3,4,4,4,4,4)
>x2=4-c(2,1,1,3,1,1,4,1,1,2,3,3)
>x3=4-c(3,3,4,1,2,4,2,2,3,1,1,2)
>x4=4-c(4,4,3,4,4,2,1,3,2,3,2,1)

# model
>model <- glm(y ~ x1+x2+x3+x4, family=poisson(link=log))
>summary(model)
```

The estimates are now

$$\tilde{\boldsymbol{\theta}} = (-0.4759, 0.7725, 0.2412, 0),$$

and with an intercept at -0.8726. Maybe a way to calculate the correct item parameters $\boldsymbol{\theta}$ from $\tilde{\boldsymbol{\theta}}$, using the intercept in some way, could be found.

Chapter 7

Models with Latent Classes

The models presented up till now, have been adequate to model responses, where consumers originate from a homogenous population. That means every consumer is assumed to have the same underlying preference scale. This assumption might be realistic for trained test panels, but when using untrained test persons this assumption is very strong. Therefor this section will be dedicated to describe models coping with panel segmentations.

The basic idea is that the heterogeneous consumer group may be partitioned into a small number of homogenious subgroups. The models are named *Latent Class* (LC) models, refering to the latent segmentation of the consumers. A segmentation which can not be identified directly, but only observed through the preference estimates.

Recall the categories of models for density estimation, presented in section 2; parametric and non-parametric models. Until now the models described in the thesis have been of the parametric kind. In this section a semi-parametric model will be used, also called mixture model. The LC ranking models are *finite mixture models*. This section will therefor provide an introduction to mixture models and how they are used within ranking problems.

The model description is followed by a presentation of an the iterative algorithm used for parameter estimation, called the *Expectation Maximization* (EM) algorithm.

7.1 Mixture Models

Semi-parametric models are models which fall somewhere in between the parametric and the non-parametric models. They allow for some flexibility, but are still limiting the number of variables. One kind of semi-parametric models are mixture models, where the density function is assumed to be a linear combination or mixture of a small number of parametric density functions. As the number of parametric density functions increase the semi-parametric model will approach a non-parametric model.

If the number of parametric density functions, which can be thought of as basis functions is fixed, the model is called a *finite* mixture model. This is the case for the LC models.

The density probability function in a mixture model can be derived using the Bayes Theorem of conditional probability. In [2] this is done, and the density function $p(x)$ is given as

$$p(x) \quad = \quad \sum_{s=1}^{m} p(x|s)P(s),$$

where $p(x|s)$ is the s'th parametric density function (or basis function), and the linear coefficients $P(s)$ for $s = 1, \ldots, m$ are called the mixing parameters or prior probabilities. The prior probabilities maintain the following conditions

$$\sum_{s=1}^{k} P(s) = 1 \qquad \wedge \qquad 0 \leq P(s) \leq 1,$$

ensuring that $p(x)$ satisfy the conditions of a density function, since of cause all the parametric density functions are probability density functions obeying the normalization condition

$$\int p(x|s)dx \quad = \quad 1.$$

The combined probability of observing x_i for $i = 1, \ldots, q$

$$\begin{aligned} P(\mathbf{X} = \mathbf{x}) &= \prod_{i=1}^{q} p(x_i) & (7.1) \\ &= \prod_{i=1}^{q} \sum_{s=1}^{m} p(x_i|s)P(s). & (7.2) \end{aligned}$$

7.2 Mixture models within Ranking Problems

Recall the ranking model presented in 6.3, where the stochastic variable of the number of times r_u is observed in the data, is

$$Y_u = \sum_{k=1}^{n} Y_{uk},$$

and the multiple stochastic variable of those

$$\mathbf{Y} = (Y_1, \ldots, Y_{t!}),$$

where all components are independent, and from which the observation $\mathbf{y} = (y_1, \ldots, y_{t!})$ has been drawn.

The total probability of the observed data, is still

$$P(\mathbf{Y} = \mathbf{y}) = \prod_{u=1}^{t!} P_{LC}(Y_u = y_u), \qquad (7.3)$$

but $P_{LC}(Y_u = y_u)$ is now modeled according to a finite mixture model, to hold the latent classes.

Define an unobserved categorical stochastic variable Z. An observation from Z identify the latent class s from which a random consumer originate. That is

$$P(Z = s) \quad = \quad \alpha_s \qquad \forall s = 1, \ldots, m,$$

where m is the number of latent classes in the model, and

$$\sum_{s=1}^{m} \alpha_s = 1 \qquad \wedge \qquad 0 \leq \alpha_s, \quad \forall s = 1 \ldots, m.$$

The probability density $P_{LC}(Y_u = y_u)$ in the latent class model is then, according to (7.1),

$$P_{LC}(Y_u = y_u) = \sum_{s=1}^{m} P(Y_u = y_u|Z = s)P(Z = s). \tag{7.4}$$

Each class is thought of as a homogenous consumer group with its own set of item parameters θ_s to be estimated. Therefore any ranking model could be used inside each class. In this thesis however the MBT model with Poisson distribution will be used, and as done in 6.3 the Bradley-Terry probability is assumed. Therefor the class conditioned probability of observing ranking r_u is described according to (6.4), indexed by class number

$$\begin{aligned} P(Y_u = y_u|Z = s) &= P_s(Y_u = y_u) \\ &= c(\theta_s)\prod_{i=1}^{t} \pi_{i,s}^{t-r_{iu}}, \end{aligned} \tag{7.5}$$

for all classes, $s = 1, \ldots, m$ where $\pi_{i,s} = \exp(\theta_{is})$ is preference score of item i in class s for all items and all classes.

Using (7.4) and (7.5) into (7.3), the total probability is given as

$$P(\mathbf{Y} = \mathbf{y}) = \prod_{u=1}^{t!}\sum_{s=1}^{m} P_s(Y_u = y_u)\alpha_s.$$

7.3 Parameter Estimation

Like in the earlier section the likelihood function is derived in order to find the ML parameters

$$L(\Theta, \alpha; \mathbf{y}) = \prod_{u=1}^{t!}\sum_{s=1}^{m} P_s(Y_u = y_u)\alpha_s,$$

where $\Theta = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$ is a matrix of item parameter vectors and $\alpha = (\alpha_1, \ldots, \alpha_m)$ is a vector of the prior probabilities for each class.

The log-likelihood function is therefore

$$\ell(\Theta, \alpha; y) = \sum_{u=1}^{t!} y_u \log\left(\sum_{s=1}^{k} \alpha_s(P_s(R_u))\right). \tag{7.6}$$

Inference in a finite mixture model must provide estimates both for the mixing parameters, here $\alpha_1, \dots, \alpha_m$ as well as for the parameters of each mixing component, here components of Θ. This is most often a highly non-linear problem, and an iterative method must be used to maximize the log-likelihood function.

Iterative Algorithms

The overall idea behind the iterative methods used is to alternate between optimizing on α and optimizing on $\boldsymbol{\theta}_s$ for each $s = 1, \dots, m$, until the change in the parameters pass some limit of wanted precision.

Different methods have been applied to the problem. In [19, chapter 8.6] a Gibbs sampler is proposed, which is a Markov Chain Montecarlo Method. The algorithm which will be used in this thesis is an Expectation Maximization (EM) algorithm, which is closely related to the Gibbs sampler. The EM algorithm was first described in 1977 by Dempster et al, and is implemented in the statistical software R, in the function `flexmix`.

Expectation Maximization Algorithm

The goal of the algorithm is to maximize the log-likelihood function (7.6), to estimate the prior probabilities $\alpha_1, \dots, \alpha_m$ as well as the item parameters $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m$ for each class.

The algorithm alternates between an Expectation step (E-step) and a Maximization step (M-step), until the change in the parameters pass the precision criteria.

An initial guess on the item parameters $\boldsymbol{\theta}_s$ in each class, must be made, together with an initial guess on the posterior probability of each observation. This is a classical starting guess problem, if nothing is known, one guess can be as good as the other. In the tests made on ranking data in this thesis, the starting guess will be done in the following way.

Divide the rankings $r_1, \dots, r_{t!}$ into m groups, where m is the number of latent classes to fit. If the rank number u tells anything about the difference in the rankings, eg. maybe rank r_u is ”near” to r_{u+1}, then this knowledge should be used. Then set the posterior probability of the observations from the rankings in group s to be $Z = s$ for all m groups.

For each group an ordinary ML estimate is found according to the ranking model used. Which in this section is the MBT with Poisson distribution and BT probability. This procedure gives the initial guess on Θ and the posterior probabilities.

Now the algorithm can be started, alternating between the two steps.

Maximization step

New estimates for Θ are found in this step, by maximizing the log-likelihood function using the probabilities $P_s(Y_u = y_u)$ and prior probability estimates $\tilde{\alpha}_s$.

For every class, $s = 1, \ldots, m$

$$\tilde{\boldsymbol{\theta}}_s \quad = \quad \text{argmax}_{\boldsymbol{\theta}_s}\{\ell(\boldsymbol{\theta}; \mathbf{y})\}.$$

Expectation step

The calculations made in this step are derivations of the expected value of the posterior weights α, to be used as an estimate of α in the next M-step.

$$\tilde{\alpha}_s \quad = \quad \frac{1}{n}\sum_{k=1}^{n} P(Z = s|Y_u = y_u),$$

where $P(Z = s|Y_u = y_u)$, the posterior probability, have been calculated using Bayes Theorem,

$$\begin{aligned} P(Z = s|Y_u = y_u) \quad &= \quad \frac{\tilde{\alpha}_s P(Y_u = y_u|Z = s)}{\sum_{s=1}^{m} \tilde{\alpha}_s P(Y_u = y_u|Z = s)} \\ &= \quad \frac{\tilde{\alpha}_s P_s(Y_u = y_u)}{\sum_{s=1}^{m} \tilde{\alpha}_s P_S(Y_u = y_u)}, \end{aligned}$$

using the "old" α estimates.

The algorithm is described by the following pseudo code.

Algorithm 1 Expectation Maximization Algorithm

```
Require: : $n$ rank observations $obs_j = \{y_{1j}, \dots, y_{t!j}\}\ \forall j = 1, \dots, n$
    $k$ = number of latent classes.
make initial guess on $\theta$ for every class, $\theta_{0s},\ \forall s = 1, \dots, k$
make initial guess on $\alpha$ for every class $\alpha_{0s},\ \forall s = 1, \dots, k$
while $\ell_{new} - \ell_{old} >$ threshold do
    Calculate $P_s(obs_j)$ for every observation $j = 1, \dots, n$
    Estimate posterior probabilities
        $\hat{p}_{js} = \frac{\alpha_s P_s(obs_j)}{\sum_{s=1}^{k} \alpha_s P_s(obs_j)}$
    Estimate new prior probabilities
        $\tilde{\alpha}_s = \frac{1}{n} \sum_{j=1}^{n} \hat{p}_{js}$
    Estimate item parameters for each class $s = 1, \dots, k$
        $\tilde{\boldsymbol{\theta}}_s = \text{argmax}_\theta \{\ell(\boldsymbol{\theta}; \mathbf{y}, \hat{p}_{js}, \tilde{\alpha}_s)\}$
    Update $\ell_{old}$ and $\ell_{new}$
end while
```

Implementations of EM

The EM algorithm for estimating parameters in a latent class ranking model, as described above, has been implemented in a MatLab file `EM.m`, which can be seen in appendix F.

Example

As an illustrative example a model with two latent classes was fitted to the Salad data presented in Tabel 6.1.

The code for the example, which also describes the initial guesses, can be found in script `testEM.m` in appendix F.

The algorithm is stopped when the the difference in the log-likelihood distributions gets below 10^{-6}. There seems to be a weakness in the implementation, since the posterior probabilities change in an odd way, when the precision limit is increased. This must be looked into before further use. To see the posterior weights see the script `EMoutput.m` in appendix F.

Another implementation of the EM algorithm is found in the R-package FlexMix, see reference [11]. The standard assumption in FlexMix is that the m multiple stochastic variable from which the data is observed has components with independent distributions of the exponentially family. Therefor each M-step in the function EM implementation `flexmix` calls `glm()` for parameter estimation. The user might implement optional M-steps, but the standard case suites fine in the case of Bradley-Terry probabilities and Poisson distribution, since it fits into the GLM framework.

The following call was made to the R function `flexmix`.

Call to flexmix

```
# DATA:  CF example 3
#========
>y=scan('latentdata.txt')
>n=32;

# Definition of factors
>salad=factor(rep(1:4,32))
>consumer=factor(rep(1:32,rep(4,32)))

>library(flexmix)

# model with 2 latent classes
>res2=flexmix(y~salad|consumer,k=2)
>summary(res2)
```

The Θ estimate, was found to be

$$\begin{aligned}\Theta &= [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2] \\ &= \begin{pmatrix} -0.9544 & -3.0000 \\ -0.5590 & -1.666 \\ 0.1163 & -1.3338 \\ 0 & 0 \end{pmatrix}\end{aligned}$$

This is however not interesting since, the number of classes m have just been chosen without reason.

7.3.1 The Number of Latent Classes

In a finite mixture model the number of mixture components, m is a constant which is assumed to be known. But the question of how to define the influential number m, has been ignored until now.

When the mixture model is a latent class model, then sometimes knowledge about the origin of the data indicates how many latent classes to seek for. If this is not the case, it might be possible to identify the number of classes by looking at the observations graphically. Maybe the observations naturally cluster into m more or less district groups.

If m is unknown, a theoretical way to estimate m must be found. This could be to estimate parameters for models with different number of classes, and then compare the model with m classes against the model with $m + 1$ classes, either starting with 1 class and then increase m, or start with the saturated model where $m = n$, and then compare it to the model with $m = n - 1$ classes etc.

Different tests for how good the model fits data could be used for these tests, for brief examples see [16, chapter 2.8], but for a more thorough review see [19, chaptor 7] on model assessment and model selection.

The more parameters the more flexible the model will be. In the saturated model even interpolation is possible. The test should therefore somehow encourage a good fit, but penalize on the number of parameters. The penalization term can only be determined from a choice of how "smooth" the solution should be.

In [16] the Aikaike's Distance Criteria (AIC), is presented as one way to measure the "goodness" of a model. AIC is also implemented in the R function `flexmix`.

In R the definition of AIC is given as

$$AIC = -2\ell + \kappa\nu, \tag{7.7}$$

where ν is the number of parameters in the model, and κ is some parameter of AIC. In the default AIC, $\kappa = 2$, but a choice like $\kappa = \log(\nu)$ is possible.

This definition means that a low AIC value indicates a better model, than a model with a high AIC value.

Using AIC distance criteria as a way to choose the number of latent classes, which is in practice just a choice of the flexibility of the model, seems a bit naive, if the parameter κ is not chosen whit respect to the present model and wanted smoothness.

Using the default AIC to choose the number of latent classes in the Salad data, gives a number of classes equal to the number of different rank observations. So the penalization term is far to small, to have any influence on the choice, unless of cause interpolation is wanted.

Chapter 8

B & O application

This section is build upon an analysis of a set of ranking data from the Danish audio- and video-producer, Band & Olufsen. The analysis were carried out by Dina Hvas Mortensen, Department of Psychology, Aarhus University, Denmark, and Søren Bech, Bang & Olufsen, Struer, Denmark and is described in [14].

A presentation of the data and the experiments from which it arrive will lead to a review of how the analysis is made in the article and which alternatives the GLM framework provides for the same tests. At last this presentation goes a bit further than the article from B&O, by analyzing the data for panel segmentations.

Design of Experiments

One of the interests of B&O, as designer of technological equipment, is to understand the perceptual experience of tool use, and how the individual senses interact in this context. A lot of equipment in all technology fields today use touch screens. To design touch screens, the visual appearance is a question, but what about the stimulation of the other senses? Would it make sense to artificially add haptic and auditive information to the user?

To understand more about what the stimulation of the three senses, sight, hearing and sense of touch, mean for a users performance of turning on a switch, Dina Mortensen arranged a test of the preference of different switches.

The items tested were 10 different switches (e.g. toggle switches, push button switches etc). The test participants were 88 employees at B&O. The task of the test participants were to rank the 10 switches by preference. Several tests were made, some with all senses available, some with only one or two senses available.

The tests were grouped in seven groups by available senses. There were 10 to 20 tests made in each group:

- Q1 Vision Only
- Q2 Audition only
- Q3 Haptic only
- Q4 vision and Audition only
- Q5 Vision and Haptic only
- Q6 Haptic and Audition only
- Q7 All senses available

For some (unknown) reason the data from all 88 consumers was reduced to only 10 observations per group. As this is an unreasonable loss of information the whole data set will be used in for the following analysis.

The practical execution of the experiments were done by "covering" the senses by a combination of the following; headphones with pink noise, black painted swimming goggles, and for the sense of touch; hockey gloves and diving gloves on top of cotton inner gloves. For more details see [14].

8.1 The data Analysis

The main research questions asked in [14] are:

1. How does the sensory modalities interact in the specific user context.

2. What sensory modalities are important in the full perceptual experience of that user context.

These questions were answered through the following analysis of data.

Are all switches equal?

To make any inference of the data, the hypothesis about all switches being equal must be rejected.

Getting a rough impression on how the switches are preferred, a spaghetti plot is made for each test.

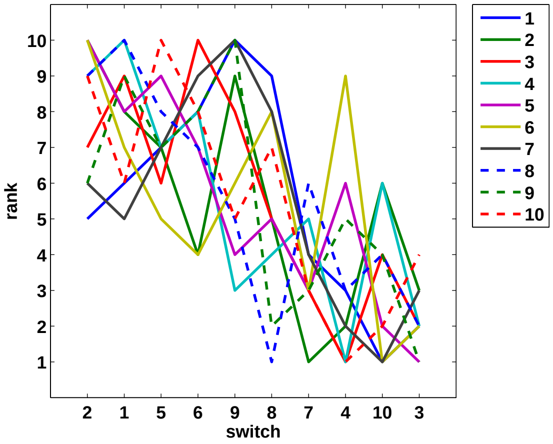


Figure 8.1: Ranks Q7

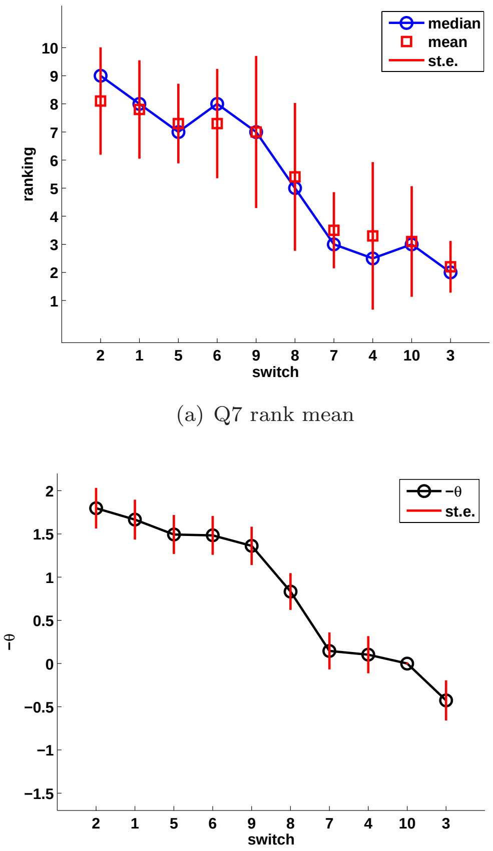
Figure 8.1 shows a trend in the rankings of the switches. The ranking of the other tests (Q1, . . . ,Q6) can be seen in appendix A. They too, give the same impression about some trend for all the tests.

B&O tested the hypothesis that all switches are preferred equal using Friedmans test. The test indicated that at least two of the switches are different with respect to preference, since the null hypothesis was rejected for all test conditions.

To support the Friedmans test a plot of the rank median and rank mean values including a 95% confidence interval, were made for each test. The results confirm that there are significant differences between the rankings of the switches. The problem about analyzing on the rank mean values are that the data is categorical and not numerical. Therefor the rank mean is only theoretical valid if it has been shown that the categories lies equidistant on the preference scale. A better alternative would be to analyze on the item parameters, estimated by a ranking model of the data.

As shown in 6, the best ranking model is the MBT. Even though the BTL will be used in the following analysis, since the MBT requires the construction of a very large model matrix, which can not be handled within R, even with a memory limit at 4 GB.

The drawback of using BTL instead of MBT was described in 6.



(a) Q7 rank mean

(b) Q7 (negative) item parameters θ

Figure 8.2: Rank mean and item parameters plotted with corresponding standard errors. The item parameters are plotted negative, to facilitate comparison of the plots.

The same plots as Figure 8.2 has been made for the other test conditions and can be found in appendix B. The plots in the appendix show, as Figure 8.2, that the rank mean gives a relative good indication of the trend in the item parameters, despite the unexplained assumption about equidistance between the categories on the preference scale.

Notice that the standard deviations on the two plots can not be compared by the size of the lines, because in the first Figure the standard deviation is calculated on the raw rank data, and in the second Figure it is calculated on the θ parameters, which are only described relative to each other.

To investigate the assumption about equidistance between the categories on the preference scale, the transformation function for the category levels are plotted.

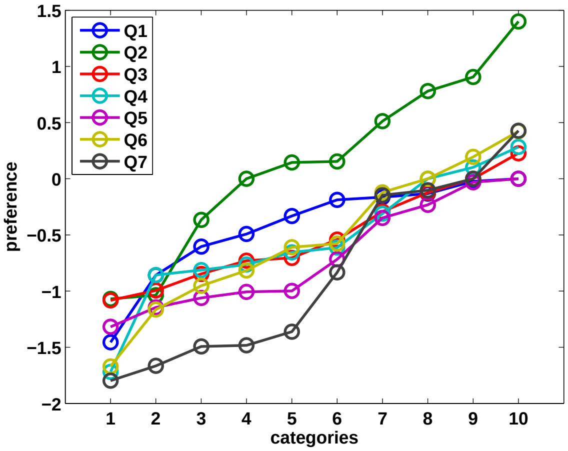


Figure 8.3: Optimal Scaling of the category levels.

Figure 8.3 show the non-linear scaling or transformation from the ranking category scale $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ to the *correct* underlying preference scale. This means that if Figure 8.3 had shown straight lines, it could be concluded that the intervals between the switch parameters $\theta_1, \ldots, \theta_{10}$ on the underlying preference scale would be equidistant. The greater the slope between two categories, the greater the distance between the categories on the preference scale.

It is seen that for some of the test conditions the intervals between the categories differ more than for others. A test on whether this is a significant result could be interesting to carry out. Together with a test on wether these graphs are significantly different from straight lines. That is wether the ratio between the item parameters could be assumed to be equal.

But before these tests have been carried out the most correct strategy must be to incorporate the non-equidistance between the category levels in the model.

Multidimensional structure in data

Through a Mann-Whitneys U-test B&O suggests that more than one underlying perceptual dimension have been applied in determining the ranking. To explore this further, Spearmans rank correlations were calculated between the test conditions.

The result of the test show that there are at least two major groups with respect to preference, one including conditions Q1, Q2 and Q4, and another including conditions Q3, Q5, Q6 and Q7. For details see [14, Table 2].

To investigate this indication of a multidimensional structure of the data, B&O conduct a Principal Component Analysis. Since the data is categorical, a Categorical Principal Component Analysis procedure (CATPCA) is used. The CATPCA quantifies categorical variables using optimal scaling. This results in optimal principal components for the transformed variables.

The eigenvalues associated with the dimensions calculated using CATPCA of the rank mean is given in Table 8.1, together with the eigenvalues associated with dimensions calculated using ordinary PCA on the θ parameters.

Dimension	CATPCA		PCA	
	of mean	[%]	of θ	[%]
1	2.146	30.66	3.5814	51.16
2	1.678	23.97	2.4264	34.66
3	0.949	13.56	0.4849	6.93
4	0.849	12.13	0.2992	4.27
5	0.626	8.94	0.0146	1.49
6	0.578	8.26	0.1045	1.27
7	0.173	2.47	0.0890	0.21

Table 8.1: The eigenvalue of the seven principal components, with CATPCA on the rank mean, and with PCA on $\boldsymbol{\theta}$.

Table 8.1 shows that both the CATPCA and the ordinary PCA on the $\boldsymbol{\theta}$ parameters indicate that the most variance can be described by two dimensions. Though the CATPCA claims that 56% of the variance could be described by two dimensions, while the PCA on $\boldsymbol{\theta}$ shows that 85% could be described by two dimensions.

Since B&O uses CATPCA they claim to model the transformation, so the results from CATPCA and PCA on $\boldsymbol{\theta}$ should be comparable. However there is a big difference in the variance described by the first two principal components, by

	Dimensions					
Test	1		2		3	
	mean	θ	mean	θ	mean	θ
Q1 Vision	-0.509	0.0021	**0.697**	**0.6100**	-0.210	0.1810
Q2 Audio	-0.123	0.1579	**0.807**	**0.5455**	-0.307	-0.0230
Q3 Tactility	**0.624**	**0.4842**	0.175	-0.0076	0.532	0.4973
Q4 Vision and Audio	-0.321	0.2725	**0.642**	**0.5062**	0.517	-0.3092
Q5 Vision and Tactility	**0.666**	**0.4740**	0.340	-0.1504	0.222	0.4277
Q6 Tactility and Audio	**0.775**	**0.4403**	0.159	-0.1605	-0.455	-0.6487
Q7 All	**0.792**	**0.4978**	0.254	-0.1601	-0.1415	-0.1415

Table 8.2: The eigenvectors of the first three dimensions, for both CATPCA on the rank mean values and PCA on the item parameters θ.

the two methods.

It is not known to the author how CATPCA perform the optimal scaling, but it is obvious that the scaling does not hold as much of the variation as the scaling done with $\boldsymbol{\theta}$. Whether this has something to do with the BTL model being used is unknown, but it could be investigated by applying the MBT model to the data, and then conduct the same PCA, if a way to deal with the size of the model matrix is found.

Anyway it must be concluded that two dimensions account for the most of the variance in the data.

What defines the dimensions?

Two dimensions has been found in the data, but what defines these dimensions? This is one of the research questions asked in [14] as *Which senses are crucial when performing the task of using a switch?*.

To find out, the eigenvectors of the first three dimensions were calculated. The results are shown in Table 8.2.

From Table 8.2 it is seen that the first dimension of the data is highly expressed by Q3, Q5, Q6 and Q7, while the second dimension is described by Q1, Q2 and Q4. An equal loading on Q3 and Q4 is observed in the third dimension for the eigenvectors calculated by CATPCA. As noticed in [14], this does not match the result of the Spearman rank correlation made. It is argued that it might be due to the transformation applied by CATPCA. But since the high correlation between Q3 and Q4 is not found when calculating on the item parameters, it must be concluded that the high correlation found, can not be due to a correct transformation.

A loading plot of the first two dimensions is made for both the CATPCA analysis on the rank means (Figure 8.4) and for the PCA on the item parameters (Figure 8.5). Both giving the information of how the test conditions are projected on to the principal components.

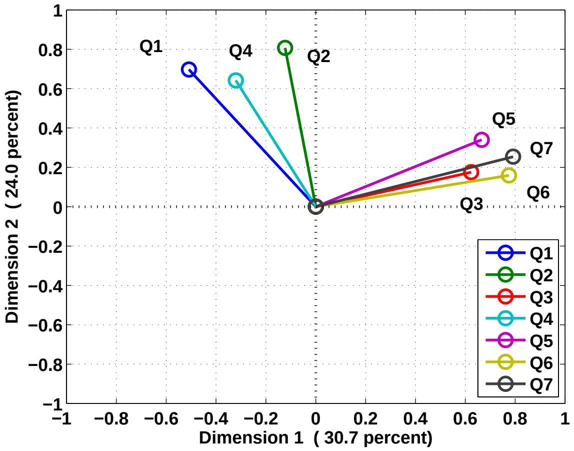


Figure 8.4: Loadingplot for CATPCA on rank mean values.

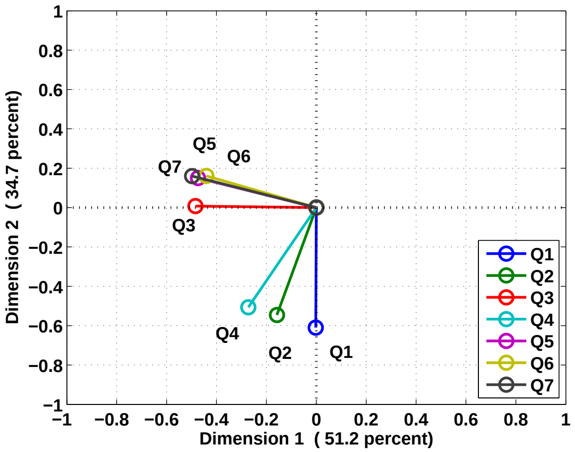


Figure 8.5: Loading plot for PCA on θ values.

Both Figure 8.4 and 8.5 shows that the haptic information plays a crucial role in how the switches are preferred. Since Q3, Q5 and Q6 are all close to Q7 (no senses covered) the plot indicates that as long as the haptic information is available there is only little difference from having full sensibility. In other words the haptic information is the most important information, even more important than a combination of the other senses.

According to [3], a score plot is the two dimensional plot, which best shows the euclidian distances. Score plots of the the estimates from the PCA analysis are shown in Figure 8.6.

8.2 Panel Segmentation

The analysis made in the article from B&O, has relied on the assumption that the preferences of each consumer are equally distributed. This is a strong assumption, for the relatively untrained panel. Therefor an analysis of panel segmentations for the Q7-tests (all senses available), will be performed in the following section. First it will be investigated wether there is a reason to believe that the consumers differ in preference when all senses are available. Since this is the case, an analysis of which consumers fall into which class will follow.

An important thing to remember about the B&O dataset, is that it only consists of 10 consumers. This is a quite small dataset and so the uncertainty of the result might be relative high. Despite this, the data will provide an instructive example of a latent class analysis.

The tests are carried out using the R package `flexmix`. The code and output can be found in appendix G.

How many Latent Classes?

To estimate the number of Latent Classes in the data, models with increasing number of classes are fitted, and the default Akaike's Information Criteria (AIC) is calculated for each model. As an example, the call and output from R is showed for the case where two latent classes are assumed.

```
> model2=flexmix(x~switch|consumer,k=2)
> summary(model2)

Call:
flexmix(formula = x ~ switch | consumer, k = 2)

       prior size post>0 ratio
Comp.1 0.301   30     80 0.375
Comp.2 0.699   70    100 0.700

’log Lik.’ -196.5404 (df=23)
AIC: 439.0809   BIC: 498.9998
```

From the output example it is seen that assuming two latent classes the AIC is 439.0809. By fitting models with consecutively increasing number of classes, the best model, according to the AIC is the model where the number of classes equals the number of consumers. The same situations as mentioned in section 7 with the Salad data.

It is therefore assumed that the best model is the model with two classes, chosen from the smoothness of the solution. The call to `flexmix` show that if two classes is fittet, the consumers will be divided into classes of 3 and 7 consumers.

Which consumer to which class?

Looking at the posterior probabilities for each consumer for each class tells how likely the observation from a given consumer is in a given class. As the chosen model only has two latent classes only the posterior probabilities for one of the classes are examined since the probabilities sum to one over all classes.

Figure 8.7 clearly indicates that the rank observation made from consumer 1, 6 and 7 is highly possible to originate from class 2, while the opposite must be valid for the rest of the observed ranks. However that the posterior probabilies split up like this, is because the problem degenerate into two distinct problems. The normal situation would be that the probability mass of the posterior was more spread out over the different classes.

Because the data is to big to be handled within the GLM framework with a Poisson distribution, an alternative solution to estimate the item parameters have been chosen. The data is split into two distinct data sets, and the item parameters are estimated within each group. This is done as an alternative to estimate the parameters correctly through the latent model, weighted by the posterior probabilities.

The observation divides the panel into two classes, class 1, consisting of consumer 2,3,4,5,8,9 and 10, and class 2 consisting of consumer 1, 6 and 7. Because of the problem by handling the data in the Poisson model, the rank mean of each class has been derived, and Figure 8.8 show the result. Even though the trend is somewhat the same for the two classes, it is easy to spot the differences too.

In stead of looking at rank mean which does not take the non-linearity of the preference scale into account, the item parameters (θ) were derived for the two classes.

The estimates were found to be

$$(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \quad = \quad \begin{pmatrix} -1.85448 & -7.355 \\ -1.88224 & -7.424 \\ 0.84368 & -7.789 \\ 0.53934 & -5.496 \\ -1.49968 & -6.915 \\ -1.35833 & -7.531 \\ 0.23201 & -6.382 \\ -0.08213 & -8.303 \\ -0.89304 & -8.432 \\ 0 & 0 \end{pmatrix},$$

by calling the R function `glm`. Both calls can be found in appendix G in the script `binprobitclasses.m`. The call and output from R for the model inference of class 2 were

Call to glm - B&O data

```
>modelclass2 <- glm(cbind(Y,n-Y) ~ X1+X2+X3+X4+X5+X6+X7+X8+X9+X10-1,
             family=binomial(link=probit), data=BOclass2)
> summary(modelclass2)

Call:
glm(formula = cbind(Y, n - Y) ~ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOclass2)

Deviance Residuals:
        Min          1Q      Median          3Q         Max
-1.387e+00  -2.692e-01  -1.366e-07   3.561e-01   1.336e+00

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1    -7.424    451.915  -0.016    0.987
X2    -7.798    451.915  -0.017    0.986
X3    -5.496    451.915  -0.012    0.990
X4    -6.915    451.915  -0.015    0.988
X5    -7.531    451.915  -0.017    0.987
X6    -7.780    451.915  -0.017    0.986
X7    -6.382    451.915  -0.014    0.989
X8    -8.303    451.915  -0.018    0.985
X9    -8.432    451.915  -0.019    0.985
X10       NA         NA      NA       NA

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 118.406  on 45  degrees of freedom
Residual deviance:  21.319  on 36  degrees of freedom
AIC: 68.513

Number of Fisher Scoring iterations: 18
```

It is seen that the $\boldsymbol{\theta}$ values vary around -7 which seems a bit strange relative to the previous tests made. The standard error however is high, and it is assumed that the strange result occur as a consequence of the small data set of only three rank observations. Never the less the same tendency for the item parameters is found for the mean rank values in Figure 8.8. This tendency shown in Figure 8.9.

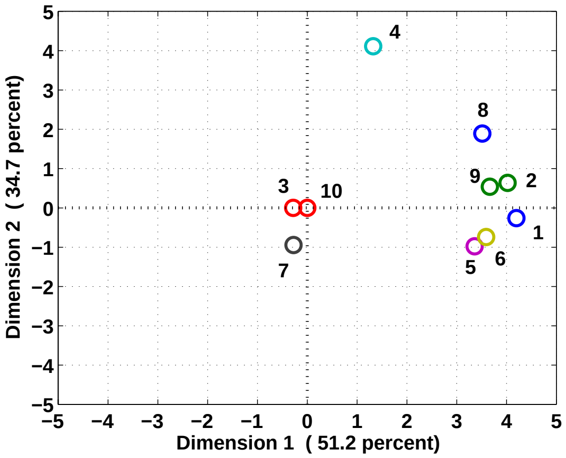


Figure 8.6: Score plot for PCA on θ values.

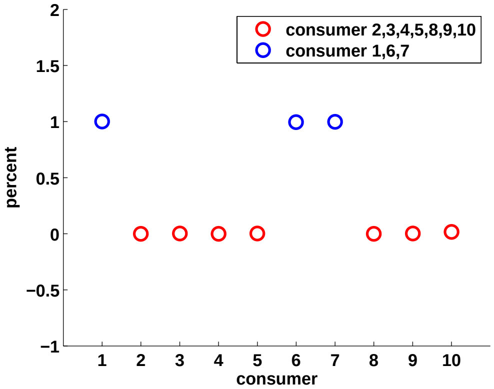


Figure 8.7: Posterior probabilities of consumer observations originating from class 2.

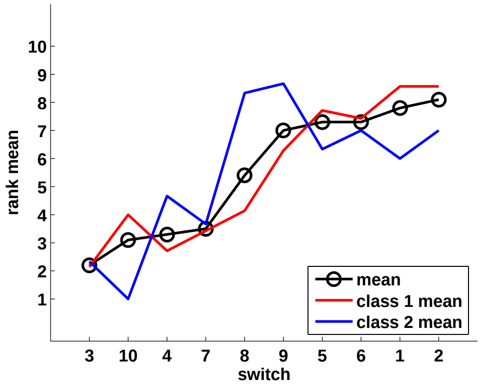


Figure 8.8: Rank mean for each latent class, plotted with the mean for the case where no panel segmentation was assumed.

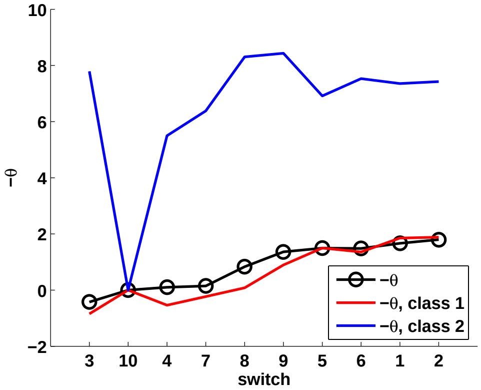


Figure 8.9: Item parameter $\boldsymbol{\theta}$ for each latent class, plotted with the item parameter for the case where no panel segmentation was assumed.

Chapter 9

Conclusion

This thesis contribute to the existing literature on the field of analysis of ranked preference data, as it provides an introductory, but mathematical stringent description of paired comparison based ranking models.

A possible mathematical background for the Thurstone and Guttman scaling models have been proposed, and it has been seen that the background is consistent with the paired comparison models, Thurstone-Mosteller and Bradley-Terry.

Ranking Models based on paired comparison models have been treated very thorough in a mathematical stringent way. Other possible ranking models have been mentioned to imply the diversity in which the models operate.

It has been shown that the MBT model and the variant of the MBT model replacing the polynomial distribution with a Poisson distribution, give the same maximum likelihood estimates. But that the Poisson version has the advantage that it can be written as a GLM.

This result give the necessary argumentation for the algorithms used for inference in [5].

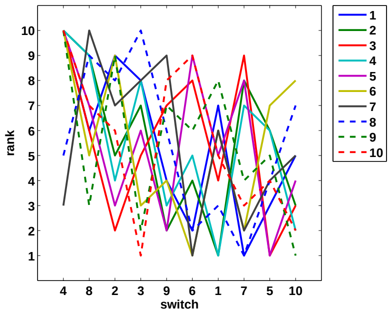
The BTL method presented in [4], as an GLM alternative to the MTB model is showed to have some simplifications inconsistent with rank data. The theoretical comparison of BTL and the MTB models which according to Philippe Courcoux has not been seen before, has been given in this thesis.

A theoretical presentation of analysis of panel segmentations have been given, as well as a practical application on ranked preference data from the Danish audio- and videoproducer, Bang & Olufsen.

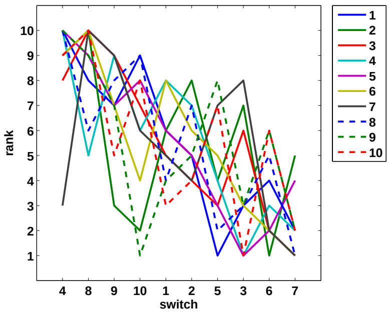
Conclusions on the analysis of the rank data from B & O was given through section 8, but a short summary could be that it was shown that the haptic information is the most important information in a job of using a switch. The haptic information is even more important than a combination of other informations.

Appendix A

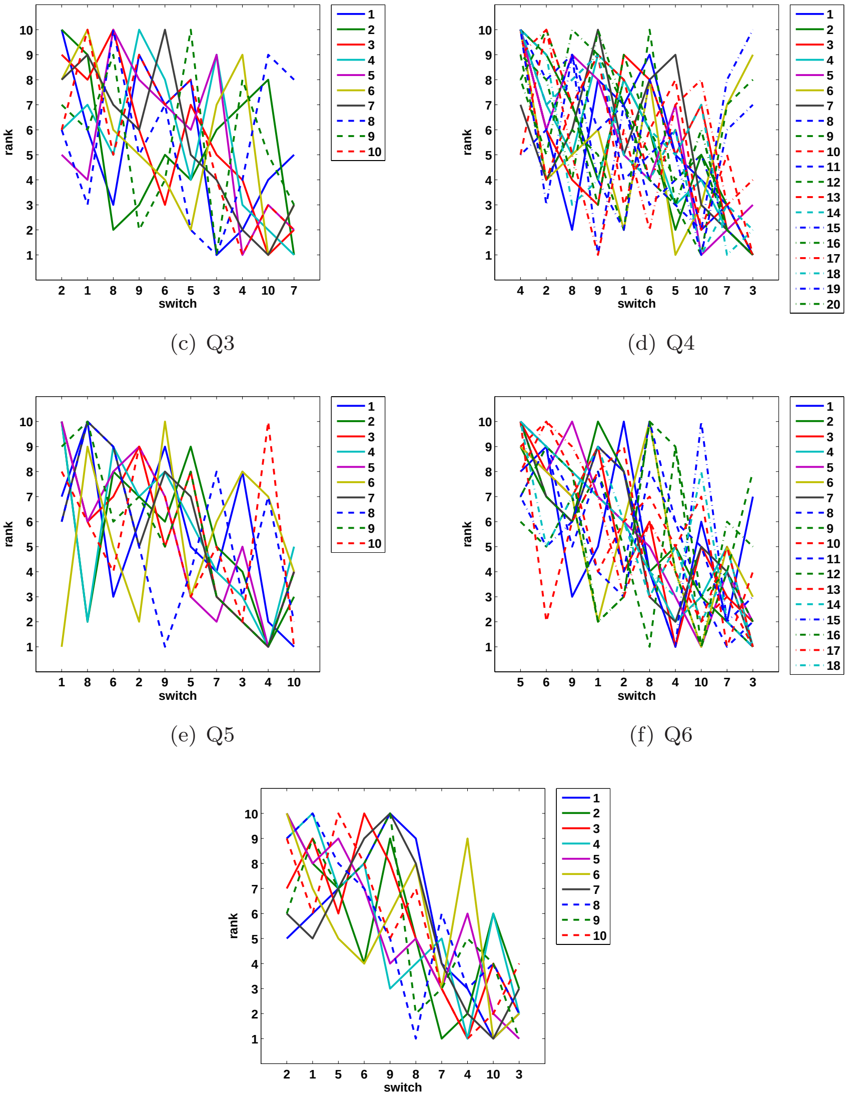
Ranks



(a) Q1



(b) Q2



(c) Q3

(d) Q4

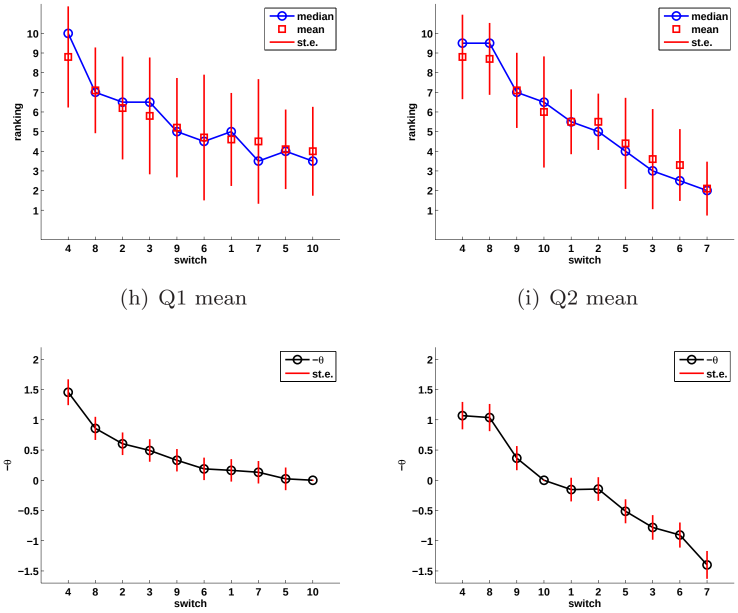
(e) Q5

(f) Q6

(g) Q7

Appendix B

Mean vs. θ

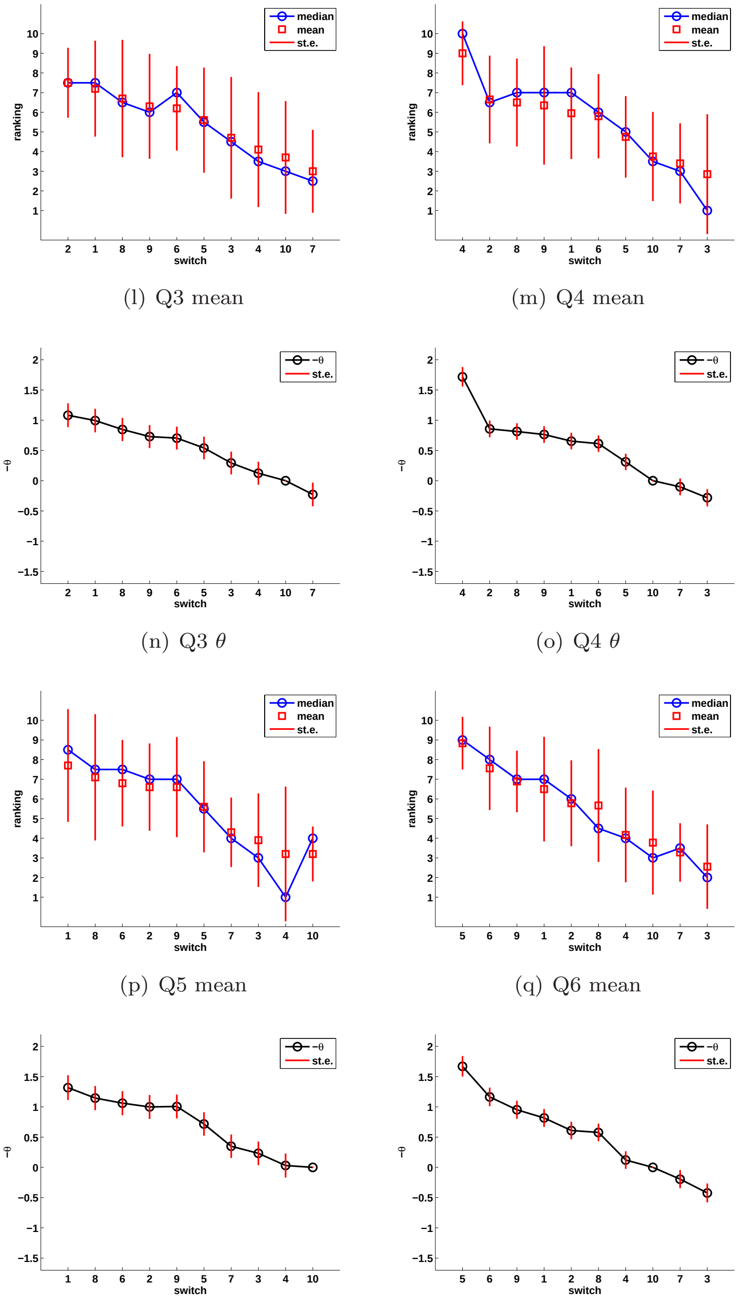


(h) Q1 mean

(i) Q2 mean

(j) Q1 θ

(k) Q2 θ



(l) Q3 mean

(m) Q4 mean

(n) Q3 θ

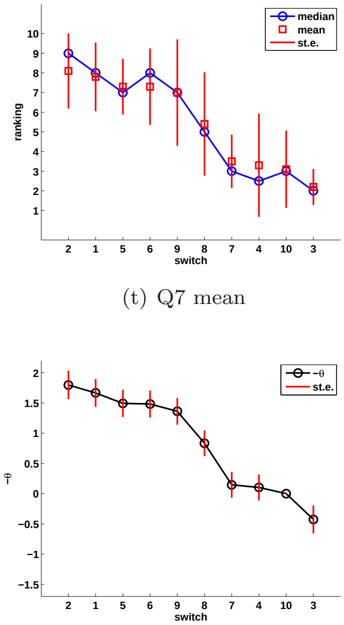
(o) Q4 θ

(p) Q5 mean

(q) Q6 mean

(r) Q5 θ

(s) Q6 θ



(t) Q7 mean

(u) Q7 θ

Appendix C

MatLab-code for ML estimation

testlikelihood.m

```
function [theta, FVAL, Exitflag] = testlikelihood(s)
% Testfunction, estimate theta parameters from optimizing on the
% loglikelihood functions. Calls direct search algorithm "fminsearch"
%
% INPUT:
% ======
% ’t’ = loglike_TM    on data from CF eks 1
% ’p’ = loglike_poiss  on data from CF eks 3
% ’b’ = loglike_BTL (call loglike TM) on data from CF eks 3
%
% OUTPUT:
% =======
% theta = parameter vector
% FVAL = function evaluation of neg. prop. loglikelihood
% Exitflag = exitflag from fminsearch

if s==’t’
% PC data from CF eks 1
Yij = [3,2,2,11,3,5]’; n=15;
theta = [1,1,1]’; items = length(theta)+1;
theta_probitCF = [-1.3874; -0.4421; -0.6192; 0];
theta_logitCF = [-2.3571; -0.7441; -1.0561; 0];
par = struct(’Yij’,Yij,’indexvector’, pairindex(items),’consumer’, n);
else
% Ranking data from CFeks 3
Yu = [2,0,0,0,0,0,1,2,1,0,0,0,2,1,0,0,0,1,11,6,3,0,1,1]’;
```

```
theta = [1,1,1]'; items = length(theta)+1;
R = ranks(items);
par = struct('Yu',Yu, 'ranks', R);
end

if s=='t' %evaluate BTL likelihood
[theta,FVAL,Exitflag] = fminsearch(@(th) loglikeTM(th,par), theta)
theta = struct('CFprobit', theta_probitCF,'CFlogit', ...
    theta_logitCF, 'MLlogit', [theta;0])

elseif s=='p' % evaluate poisslikelihood
[theta,FVAL,Exitflag] = fminsearch(@(th) loglikepoiss(th,par), theta)

elseif s=='b' % evaluate BTLlikelihood (BT paired comp)
[theta,FVAL,Exitflag] = fminsearch(@(th) loglikeBTL(th,par), theta)
end
```

loglikeTM.m

```
function LL = loglikeTM(theta, par)
% negative Loglikelihood function for the Paired Comparison, BT model
% INPUT:
% ======
% par = struct with y and indexvector.
%       y = obs-vector t(t-1)/2 (all PC)
%       theta = parametervector, length t-1 (except last)
%
% OUTPUT:
% =======
% LL = neg. functionvalue of prop. loglikelihood

% kristine@frisenfeldt.dk, 2007

theta = [theta;0];
t = length(theta);    %# of items

Yij = par.Yij;          % t(t-1)/2 x 1 -vektor
n = par.consumer;     % # consumers
index = par.indexvector;

thetai = theta(index(:,1));
thetaj = theta(index(:,2));
term1 = sum(Yij.*(thetai-thetaj));% 1x(t(t-1)/2),(t(t-1)/2)x1

pivec = exp(theta);         %[pi_1, pi_2, ... ,pi_t]'
pii = pivec(index(:,1));    %[pi_1, pi_1, p1_1, pi_2, pi_2, pi_3]'
pij = pivec(index(:,2));    %[pi_2, pi_3, p1_4, pi_3, pi_4, pi_4]'
term2 = n*sum(log((pij./(pii+pij))));

% proportional loglikelihood
LL = -sum(term1+term2)
end
```

loglikepoiss.m

```
function LL = loglikepoiss(theta, par)
% negative Loglikelihood function for the Ranking model,
% MBT with Poisson distribution
%
% INPUT:
% ======
% par = struct with y and indexvector.
%       Yu = obs-vector length t! (all ranks)
%       theta = parametervector, length t (all items)
%
% OUTPUT:
% =======
% LL = neg. functionvalue of prop. loglikelihood

% kristine@frisenfeldt.dk, 2007
y = par.Yu;          %t! x 1 -vektor
n = sum(y);         %# of consumers
theta = [theta;0];
t = length(theta);  %# of items
R = par.ranks;

c1 = normconst(theta,R);  %normalization constant
term1 = n*log(c1);
term2 = (y')*(t-R)*theta; % 1xt!, t!xt, tx1

% negative proportional loglikelihood
LL = -sum(term1+term2);

end

```

loglikeBTL.m

```
function LL = loglikeBTL(theta, par)
% negative Loglikelihood function for the BTL model
% INPUT:
% ======
% par = struct with y and indexvector.
%       Yu = obs-vector length = t! (all ranks)
%       theta = parametervector, length t-1 (except last)
%
% OUTPUT:
% =======
% LL = neg. functionvalue of prop. loglikelihood

% kristine@frisenfeldt.dk, 2007

items = length(theta)+1;
consumer = sum(par.Yu);
txtflag = 0;
% convert Ranking data to PC data
Yij = rank2pair(par.Yu, par.ranks, items, consumer, txtflag);

par = struct('Yij',Yij, 'indexvector', pairindex(items), ...
    ’consumer’ ,consumer);

% call Thurstone-Mosteller PC model
LL = loglikeTM(theta,par)
end
```

normconst.m

```
function const = normconst(theta, R)
% Calculate normalization constant c(theta)
%
% INPUT:
% ======
% theta = parameter vector (or matrix) (remember theta(t) == 0)
%
% OUTPUT:
% =======
% const = normalization constant assuring that sum_u(P(R_u))=1.
%         size(const) = 1xm (m = #latent classes)

% kristine@frisenfeldt.dk, 2007

t = size(theta,1);      %# of items
m = size(theta,2);      %# of latent classes
pivec = exp(theta);     %pi_matrix items x classes

tfak = size(R,1);       %# of ranks t!
const = [];
for s=1:m
    next = (sum(prod(repmat(pivec(:,s)’,tfak,1).^(t-R),2)))^(-1);
    const = [const,next];
end
end
```

Appendix D

R-code for GLM PC models

binprobitlogitCFeks1.R

```
# PC models (GLM - binomial distribution)
# Thurstone-Mosteller (probit)
# Bradley-Terry (logit)

# DATA: CF eksempel 1
#======
y = c(3,2,2,11,3,5)
n=rep(15,6)
x1=c(1,1,1,0,0,0)
x2=c(-1,0,0,1,1,0)
x3=c(0,-1,0,-1,0,1)
x4=c(0,0,-1,0,-1,-1)

#Thurstone-Mosteller model (probit)
model <- glm(cbind(y,n-y) ~ x1+x2+x3+x4-1, family=binomial(link=probit))
summary(model)

#OUTPUT:
#=======

Call:
glm(formula = cbind(y, n - y) ~ x1 + x2 + x3 + x4 - 1,
        family = binomial(link = probit))

Deviance Residuals:
      1        2        3        4        5        6
 0.2789  -0.8685   0.6595   1.3098  -1.1124   0.5592
```

```
Coefficients: (1 not defined because of singularities)
   Estimate Std. Error z value Pr(>|z|)
x1  -1.3874     0.2833  -4.897 9.71e-07 ***
x2  -0.4421     0.2494  -1.772   0.0763 .
x3  -0.6192     0.2516  -2.461   0.0139 *
x4       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.6890  on 6  degrees of freedom
Residual deviance:  4.5327  on 3  degrees of freedom
AIC: 27.061

Number of Fisher Scoring iterations: 4


#Bradley-Terry  model (logit)
model <- glm(cbind(y,n-y) ~ x1+x2+x3+x4-1, family=binomial(link=logit))
summary(model)

#OUPUT:
#======

Call:
glm(formula = cbind(y, n - y) ~ x1 + x2 + x3 + x4 - 1,
        family = binomial(link = logit))

Deviance Residuals:
      1        2        3        4        5        6
 0.3433  -0.8047   0.6020   1.2555  -1.0562   0.6481

Coefficients: (1 not defined because of singularities)
   Estimate Std. Error z value Pr(>|z|)
x1  -2.3571     0.5123  -4.601 4.21e-06 ***
x2  -0.7441     0.4208  -1.768   0.0771 .
x3  -1.0561     0.4290  -2.462   0.0138 *
x4       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.6890  on 6  degrees of freedom
Residual deviance:  4.2399  on 3  degrees of freedom
AIC: 26.768

Number of Fisher Scoring iterations: 4
```

Appendix E

R-code for GLM Ranking models

poisslogCFeks3.R

```
# CF eksempel 3
# Ranking model Mallows-Bradley-Terry
# GLM Poisson distribution, log-link

# DATA:  CF example 3
#========
y=c(2,0,0,0,0,0,1,2,1,0,0,0,2,1,0,0,0,1,11,6,3,0,1,1)
x1=4-c(1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4)
x2=4-c(2,2,3,3,4,4,1,1,3,3,4,4,1,1,2,2,4,4,1,1,2,2,3,3)
x3=4-c(3,4,2,4,2,3,3,4,1,4,1,3,2,4,1,4,1,2,2,3,1,3,1,2)
x4=4-c(4,3,4,2,3,2,4,3,4,1,3,1,4,2,4,1,2,1,3,2,3,1,2,1)

# model
model <- glm(y ˜ x1+x2+x3+x4, family=poisson(link=log))
summary(model)

#OUTPUT:
#=========

Call:
glm(formula = y ˜ x1 + x2 + x3 + x4, family = poisson(link = log))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9410  -0.6043  -0.3296   0.3676   2.0695

```

```
Coefficients: (1 not defined because of singularities)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.3097     1.0390  -2.223   0.0262 *
x1           -0.5548     0.2491  -2.228   0.0259 *
x2            1.1825     0.2866   4.127 3.68e-05 ***
x3            0.3776     0.2228   1.694   0.0902 .
x4                NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 70.753  on 23  degrees of freedom
Residual deviance: 22.249  on 20  degrees of freedom
AIC: 60.99

Number of Fisher Scoring iterations: 5

# DATA:  CF example 3 (-without null-terms...)
y=c(2,1,2,1,2,1,1,11,6,3,1,1)
x1=4-c(1,2,2,2,3,3,3,4,4,4,4,4)
x2=4-c(2,1,1,3,1,1,4,1,1,2,3,3)
x3=4-c(3,3,4,1,2,4,2,2,3,1,1,2)
x4=4-c(4,4,3,4,4,2,1,3,2,3,2,1)

# model
model <- glm(y ˜ x1+x2+x3+x4, family=poisson(link=log))
summary(model)

#OUTPUT:
#=========

Call:
glm(formula = y ˜ x1 + x2 + x3 + x4, family = poisson(link = log))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.22657  -0.73067  -0.07679   0.42284   1.44781

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.8726     1.1196  -0.779  0.43575
x1           -0.4759     0.2526  -1.884  0.05961 .
x2            0.7725     0.2875   2.687  0.00722 **
x3            0.2412     0.2564   0.941  0.34695
x4                NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 26.3912  on 11  degrees of freedom
Residual deviance:  9.4115  on  8  degrees of freedom
AIC: 48.153
```

```
Number of Fisher Scoring iterations: 5
```

binprobitlogitCFeks3.R

```
# CF eksempel 3
# Converted Ranking data to PC data
# Ranking model BTL
# GLM binomial distribution, logit-link
# (also version with probit-link)

# DATA:
#======
y = c(4,6,8,25,29,21)
n=rep(32,6)
x1=c(1,1,1,0,0,0)
x2=c(-1,0,0,1,1,0)
x3=c(0,-1,0,-1,0,1)
x4=c(0,0,-1,0,-1,-1)

#Thurstone-Mosteller model (probit)
model <- glm(cbind(y,n-y) ~ x1+x2+x3+x4-1, family=binomial(link=probit))
summary(model)

#OUTPUT:
#=======

Call:
glm(formula = cbind(y, n - y) ~ x1 + x2 + x3 + x4 - 1,
        family = binomial(link = probit))

Deviance Residuals:
      1        2        3        4        5        6
 1.3136  -0.1734  -0.8414   0.3245   0.8466   0.1464

Coefficients: (1 not defined because of singularities)
   Estimate Std. Error z value Pr(>|z|)
x1  -0.4740     0.1753  -2.704  0.00686 **
x2   1.0654     0.1879   5.669 1.43e-08 ***
x3   0.3689     0.1679   2.197  0.02804 *
x4       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 80.465  on 6  degrees of freedom
Residual deviance:  3.307  on 3  degrees of freedom
AIC: 29.832

Number of Fisher Scoring iterations: 4


#Bradley-Terry  model (logit)
model <- glm(cbind(y,n-y) ~ x1+x2+x3+x4-1, family=binomial(link=logit))
summary(model)
```

```
#OUPUT:
#======

Call:
glm(formula = cbind(y, n - y) ~ x1 + x2 + x3 + x4 - 1,
        family = binomial(link = logit))

Deviance Residuals:
       1         2         3         4         5         6
 1.19370  -0.06114  -0.67964   0.16454   0.79619   0.09420

Coefficients: (1 not defined because of singularities)
   Estimate Std. Error z value Pr(>|z|)
x1  -0.8271     0.3009  -2.749  0.00598 **
x2   1.8147     0.3391   5.352 8.69e-08 ***
x3   0.6116     0.2829   2.162  0.03059 *
x4       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 80.4645  on 6  degrees of freedom
Residual deviance:  2.5604  on 3  degrees of freedom
AIC: 29.085

Number of Fisher Scoring iterations: 4
```

Appendix F

MatLab-code for EM implementation

testEM.m

```
% This script tests the EM algorithm implementet in EM.m
% The data is the Ranking Data from CF eks 3, with 2 classes
%
% Initial guesses are made, and then the funciton is called.


Yu = [2,0,0,0,0,0,1,2,1,0,0,0,2,1,0,0,0,1,11,6,3,0,1,1]';
k = 2;  %# latent classes
theta0 = [1,1,1]';
items = length(theta0)+1;
R = ranks(items);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% make start guess on posterior and theta_s
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
split = round(length(Yu)/k);
Ysplit = zeros(length(Yu),k);
for s=1:k
   if s==k
       indexs = ((split*(s-1)+1):length(Yu));
       Ysplit(indexs,s) = Yu(indexs);
   else
       indexs = (split*(s-1)+1):(split*s);
       Ysplit(indexs,s) = Yu(indexs);
   end
```

```
end
theta=[];
for s=1:k
   par = struct(’Yu’,Ysplit(:,s), ’ranks’, R);
   thetas = fminsearch(@(th) loglikepoiss(th,par), theta0);
   theta = [theta,thetas];
end
theta_start = theta  %start guess
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% prior probabilities (alpha)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alpha_start = repmat(1/k,k,1); %start guess
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function call
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
par = struct(’y’,Yu, ’ranks’, R, ’theta’, theta_start, ...
    ’prior’, alpha_start);
[theta, posterior] = EM(par)
```

EM.m

```
function [thetaout, postout]= EM(par)
% Expectation Maximization Algorithm
% call loglike_latent
%
% Input:
% ========
% par = struct(’y’,Yu, ’ranks’, R, ’theta’, theta_start, ...
%                                           ... ’prior’, alpha_start);
% Output:
% ========
% thetaout = matrix with item parameter vectors for each class
% postout  = matrix with posterior prob for each observation for each class

% Kristine Frisenfeldt, 2007
R       = par.ranks;        %rankings
yu      = par.y;            %observed data
indexyj = find(yu~=0)
tfak    = size(R,1)         %t!
k       = size(par.theta,2);%# of latent classes s=1...k
t       = size(R,2);        %# of items i=1...t
n       = sum(yu);          %# of observations =# of consumers j=1...n

Lold = 0; Lnew = 1; % likelihood function values
it = 0;
while (abs(Lold-Lnew)>10e-5)
    it = it+1; Lold = Lnew

%%%%%%%%%%%%%%%%%%%% E-step  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
theta   = [par.theta;zeros(1,k)];   %txk
prior   = par.prior;                %kx1
c       = normconst(theta,R);       %1xk

% P(Yj=1|s) = PsRj,  nxk
PsRu = repmat(c,tfak,1).*exp((t-R)*theta);
PsYj = [];
for j=indexyj'
PsYj = [PsYj;repmat(PsRu(j,:),yu(j),1)];
end

% P(s|Yj=1) = posterior , kxn
post_new=(PsYj*diag(prior))./(repmat(PsYj*prior,1,k));
                                                %nxk * kxk ./ nxk * kx1

% P(s) = prior, notice sum(prior)==1
prior_new = sum(post_new)/n;
%%%%%%%%%%%%%%%%%%% end of E-step %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

par.post  = post_new;
par.prior = prior_new';

%%%%%%%%%%%%%%%%%%% M-step %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[theta_new,Lnew,Exitflag] = fminsearch(@(th) loglikelatent(th,par), ...
    par.theta);
%%%%%%%%%%%%%%%%%%% end of M-step %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

par.theta = theta_new;

end  %end forloop
thetaout = par.theta; postout =  par.post;
par.it = it; %#iterations
end %end EM-function
```

loglikelatent.m

```
function LE = loglikelatent(theta, par)
% Calculate the negative prop. loglikelihood of the latent model.
%
% INPUT:
% ======
% theta = start guess for parameters
% par   = struct with
%       y       = obs-vector length t! (all ranks)
%       R       = rank matrix
%       theta   = last parameter estimate
%       post    = last posterior estimate
%       prior   = last prior estimate
%
% OUTPUT:
% ======
% LE = negative proportional loglikelihood evaluation
% kristine@frisenfeldt.dk, 2007
R       = par.ranks;
t       = size(R,2);            %# of items
k       = size(par.theta,2);    %# of latent classes
theta   = [theta;zeros(1,k)];
cc      = normconst(theta,R);
post    = par.post;
y       = par.y;
n       = sum(y);
indexyj = find(y~=0);

temp = (t-R)*theta;      % t!xt * txk
temp1 = [];              % becomes nxk
for j=indexyj’
    temp1 = [temp1;repmat(temp(j,:),y(j),1)];
end

LE = -sum(sum((post.*(repmat(log(cc),n,1) + temp1))));

end
```

Appendix G

Code Analysis of B&O Data

binomialOutput.txt

```
# The data analysed comes from these calls
# GLM with binomial distribution and probit link.


#Q1:
#=====================================================================

Call:
glm(formula = cbind(Y, n - Y) ~ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.46990  -0.53518  -0.02375   0.47237   1.29842

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1  -0.16363    0.18614  -0.879  0.37935
X2  -0.60374    0.18733  -3.223  0.00127 **
X3  -0.49268    0.18635  -2.644  0.00820 **
X4  -1.45609    0.21374  -6.812 9.60e-12 ***
X5  -0.02331    0.18731  -0.124  0.90096
X6  -0.18810    0.18601  -1.011  0.31191
X7  -0.13239    0.18633  -0.711  0.47736
X8  -0.85807    0.19139  -4.483 7.35e-06 ***
X9  -0.33109    0.18574  -1.783  0.07466 .
```

```
X10       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 111.420  on 45  degrees of freedom
Residual deviance:  19.522  on 36  degrees of freedom
AIC: 149.01

Number of Fisher Scoring iterations: 4

#Q2:
#======================================================================

Call:
glm(formula = cbind(Y, n - Y) ˜ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.50039  -0.65690   0.01795   0.38586   1.55450

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1    0.1540     0.1958   0.787 0.431424
X2    0.1452     0.1958   0.741 0.458391
X3    0.7804     0.2040   3.825 0.000131 ***
X4   -1.0690     0.2269  -4.712 2.46e-06 ***
X5    0.5131     0.1987   2.583 0.009809 **
X6    0.9061     0.2076   4.364 1.28e-05 ***
X7    1.3996     0.2300   6.086 1.16e-09 ***
X8   -1.0368     0.2250  -4.608 4.06e-06 ***
X9   -0.3658     0.2003  -1.826 0.067800 .
X10       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 243.799  on 45  degrees of freedom
Residual deviance:  23.910  on 36  degrees of freedom
AIC: 130.31

Number of Fisher Scoring iterations: 5

#Q3:
#======================================================================
Call:
glm(formula = cbind(Y, n - Y) ˜ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.03873  -0.25317   0.01202   0.26389   0.95210

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1   -0.9941     0.1943  -5.116 3.12e-07 ***
X2   -1.0824     0.1964  -5.510 3.60e-08 ***
X3   -0.2924     0.1882  -1.554 0.120234
X4   -0.1237     0.1894  -0.653 0.513820
X5   -0.5415     0.1883  -2.875 0.004035 **
X6   -0.7046     0.1896  -3.716 0.000202 ***
X7    0.2274     0.1961   1.160 0.246232
X8   -0.8471     0.1915  -4.423 9.73e-06 ***
X9   -0.7296     0.1899  -3.842 0.000122 ***
X10       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 101.797  on 45  degrees of freedom
Residual deviance:   8.451  on 36  degrees of freedom
AIC: 141.27

Number of Fisher Scoring iterations: 4

#Q4:
#======================================================================

Call:
glm(formula = cbind(Y, n - Y) ~ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7620  -0.4921   0.1758   0.4502   1.2299

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1   -0.6546     0.1362  -4.805 1.55e-06 ***
X2   -0.8568     0.1382  -6.202 5.58e-10 ***
X3    0.2823     0.1418   1.991   0.0465 *
X4   -1.7177     0.1614 -10.642  < 2e-16 ***
X5   -0.3100     0.1354  -2.290   0.0220 *
X6   -0.6126     0.1360  -4.505 6.63e-06 ***
X7    0.1020     0.1386   0.736   0.4618
X8   -0.8127     0.1376  -5.904 3.54e-09 ***
X9   -0.7635     0.1371  -5.567 2.59e-08 ***
X10       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 305.474  on 45  degrees of freedom
Residual deviance:  25.836  on 36  degrees of freedom
AIC: 179.66

Number of Fisher Scoring iterations: 4

#Q5:
#======================================================================

Call:
glm(formula = cbind(Y, n - Y) ~ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.76511  -0.41267  -0.05444   0.43675   1.76682

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1  -1.31849    0.20451  -6.447 1.14e-10 ***
X2  -0.99860    0.19740  -5.059 4.22e-07 ***
X3  -0.23153    0.19527  -1.186 0.235748
X4  -0.03076    0.19844  -0.155 0.876810
X5  -0.71746    0.19425  -3.694 0.000221 ***
X6  -1.06141    0.19848  -5.348 8.91e-08 ***
X7  -0.34884    0.19422  -1.796 0.072485 .
X8  -1.14535    0.20015  -5.722 1.05e-08 ***
X9  -1.00651    0.19753  -5.096 3.48e-07 ***
X10       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 136.775  on 45  degrees of freedom
Residual deviance:  23.212  on 36  degrees of freedom
AIC: 150.96

Number of Fisher Scoring iterations: 5

#Q6:
#======================================================================

Call:
glm(formula = cbind(Y, n - Y) ~ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.81069  -0.47220  -0.02250   0.53442   2.20041

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
```

```
X1   -0.8172     0.1473  -5.547 2.91e-08 ***
X2   -0.6093     0.1455  -4.189 2.81e-05 ***
X3    0.4225     0.1547   2.731  0.00631 **
X4   -0.1207     0.1456  -0.829  0.40699
X5   -1.6703     0.1694  -9.858  < 2e-16 ***
X6   -1.1648     0.1532  -7.604 2.87e-14 ***
X7    0.1943     0.1495   1.300  0.19355
X8   -0.5778     0.1453  -3.977 6.98e-05 ***
X9   -0.9519     0.1492  -6.381 1.76e-10 ***
X10       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 331.449  on 45  degrees of freedom
Residual deviance:  25.684  on 36  degrees of freedom
AIC: 167.99

Number of Fisher Scoring iterations: 5

#Q7:
#======================================================================
Call:
glm(formula = cbind(Y, n - Y) ~ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3781  -0.8048  -0.1770   0.3225   1.4276

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1   -1.6657     0.2307  -7.222 5.14e-13 ***
X2   -1.7973     0.2355  -7.633 2.30e-14 ***
X3    0.4268     0.2322   1.838    0.066 .
X4   -0.1020     0.2149  -0.475    0.635
X5   -1.4933     0.2253  -6.629 3.38e-11 ***
X6   -1.4833     0.2250  -6.593 4.32e-11 ***
X7   -0.1461     0.2141  -0.682    0.495
X8   -0.8337     0.2127  -3.919 8.89e-05 ***
X9   -1.3615     0.2218  -6.139 8.30e-10 ***
X10       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 250.380  on 45  degrees of freedom
Residual deviance:  25.131  on 36  degrees of freedom
AIC: 124.76

Number of Fisher Scoring iterations: 5
```

PCA.m

```
clear all; close all;
%for title, label, legend
set(0,'defaultaxesfontsize',13, 'defaultaxesfontweight','bold');
%for line and marker
set(0,'DefaultLineLineWidth',2,'DefaultLineMarkerSize',10 );
%text ??
set(0,'defaulttextfontsize',13, 'defaulttextfontweight','bold');

% data fra GLM probit model [7sp?rgsm?l x 10knapper]
data = [-0.16362926, -0.60373591, -0.49267676, -1.45609437, ...
    -0.02330957, -0.18809978, -0.13239480, -0.85807253, -0.33108952;
 0.1540293,  0.1451606,  0.7803966, -1.0690160,  0.5130524,  ...
 0.9060717,  1.3996335, -1.0367837, -0.3657888;
-0.9941252, -1.0823540, -0.2924012, -0.1236812, -0.5414547, ...
-0.7046049,  0.2273817, -0.8470542, -0.7295774;
-0.6546101, -0.8568345 , 0.2823199, -1.7177144, -0.3100414, ...
-0.6126236,  0.1019870, -0.8127262, -0.7634603;
-1.31849228, -0.99860009, -0.23152731, -0.03076158, -0.71746365, ...
-1.06140544, -0.34883781, -1.14535492, -1.00651317;
-0.8172346, -0.6092874,  0.4224775, -0.1207263, -1.6702825, ...
-1.1648141,  0.1943458, -0.5777694, -0.9519208;
-1.6657386, -1.7972956 , 0.4268400, -0.1019803, -1.4933180, ...
-1.4832802, -0.1461087, -0.8337355, -1.3615162] ;
Q10 = zeros(7,1);
data = [data,Q10];
X = data'; %nxt %THETA DATA

%MEAN DATA
XMEAN = [];
for i = 1:7
clear Xrank; clear permut;
X = load(sprintf('Q%d.txt',i));
items = 10;
consumer = length(X)/items;
Xrank = reshape(X, items, consumer)';
XMEAN = [XMEAN,mean(Xrank,1)'];
end

X=XMEAN; %%%%%%%%%%%%%%%NOTICE!!

n = 10;
Xstreg = ((1/n)*sum(X,1))';

% Dispersion matrix S
S = (1/(n-1))*(X'*X-n*Xstreg*Xstreg')

Sd = diag(S);
sqsd = sqrt(Sd);
X=X./repmat(sqsd',10,1);
Xstreg = ((1/n)*sum(X,1))';

% Correlation matrix
R = (1/(n-1))*(X'*X-n*Xstreg*Xstreg')
```

```
% Eigenvectors and Eigenvalues
[V,D] = eig(R)

d = diag(D);
[d,I]=sort(d,1,’descend’); %sort eigenvalues, I = index
P = V(:,I); %eigenvectors, sorted.

% Accumulated percent of total varians
profvar=[];
for i = 1:length(d)
profvar = [profvar; sum(d(1:i))/sum(d)];
end

% denne figur svarer til figur 2 i B&O-paper.
clf
sorted = sort(data’,1);
plot(sorted, ’o-’);
legend(’Q1’,’Q2’,’Q3’,’Q4’,’Q5’,’Q6’,’Q7’, ’Location’, ’NorthWest’);
set(gca,’xtick’,linspace(1,10,10), ’XLim’,[0,11]);
%title(’B&O, optimal scaling, \theta’, ’fontsize’, 15);
xlabel(’categories’);
ylabel(’preference’);

% Eigenvalue table ref{}
eigcat = [2.146,1.678,0.949,0.849,0.626,0.578,0.173]’;
profvarcat = eigcat/7
eigpca = d;
profvarpca = d/7;

% Loading plots
%x = dim1, y = dim2
Vcat = [-0.509,0.697;
        -0.123,0.807;
         0.624,0.175;
         -0.321,0.642;
         0.666,0.340;
         0.775, 0.159;
         0.792, 0.254]’;
figure
plot([zeros(1,7);Vcat(1,:)],[zeros(1,7);Vcat(2,:)],’o-’); hold on
plot([-1,0;1,0],[0,-1;0,1],’:k’);
AXIS([-1 1 -1 1]);
set(gca, ’XTick’, linspace(-1,1,11), ’Ytick’, linspace(-1,1,11), ...
    ’Xgrid’, ’on’, ’Ygrid’, ’on’)
xlabel(sprintf(’Dimension 1  ( %2.1f percent)’,profvarcat(1)*100));
ylabel(sprintf(’Dimension 2  ( %2.1f percent)’,profvarcat(2)*100));
legend(’Q1’, ’Q2’, ’Q3’, ’Q4’, ’Q5’, ’Q6’, ’Q7’,’Location’, ’SouthEast’);
gtext(’Q1’);gtext(’Q2’);gtext(’Q3’);gtext(’Q4’);gtext(’Q5’);
gtext(’Q6’);gtext(’Q7’);

Vpca = V(:,1:2)’; %bem?rk negativ!!
figure
plot([zeros(1,7);Vpca(1,:)],[zeros(1,7);Vpca(2,:)],’o-’); hold on;
plot([-1,0;1,0],[0,-1;0,1],’:k’);
AXIS([-1 1 -1 1])
```

```
set(gca, ’XTick’, linspace(-1,1,11), ’Ytick’, linspace(-1,1,11), ...
    ’Xgrid’, ’on’, ’Ygrid’, ’on’)
xlabel(sprintf(’Dimension 1  ( %2.1f percent)’,profvarpca(1)*100));
ylabel(sprintf(’Dimension 2  ( %2.1f percent)’,profvarpca(2)*100));
legend(’Q1’, ’Q2’, ’Q3’, ’Q4’, ’Q5’, ’Q6’, ’Q7’, ’Location’, ’SouthEast’);
gtext(’Q1’);gtext(’Q2’);gtext(’Q3’);gtext(’Q4’);gtext(’Q5’);
gtext(’Q6’);gtext(’Q7’);

% Scoreplots
%SC = X*Vpca’; % nxp , px2
SC = X*Vcat’; % nxp , px2
figure
plot([SC(:,1),SC(:,1)]’, [SC(:,2),SC(:,2)]’,’o’); hold on;
max = 14
plot([-max,0;max,0],[0,-max;0,max],’:k’);
AXIS([-max max -max max])
set(gca, ’XTick’, linspace(-max,max,11), ’Ytick’, ...
    linspace(-max,max,11), ’Xgrid’, ’on’, ’Ygrid’, ’on’)
xlabel(sprintf(’Dimension 1  ( %2.1f percent)’,profvarpca(1)*100));
ylabel(sprintf(’Dimension 2  ( %2.1f percent)’,profvarpca(2)*100));
gtext(’1’);gtext(’2’);gtext(’3’);gtext(’4’);gtext(’5’);gtext(’6’);
gtext(’7’);gtext(’8’);gtext(’9’);gtext(’10’);
```

Q7analysis.txt

```
# Reading data
>x=scan("Q7.txt")

# Definition of factors
>switch=factor(rep(1:10,10))
>consumer=factor(rep(1:10,rep(10,10)))

>library(flexmix)

>res1=flexmix(x~switch|consumer,k=1)
>summary(res1)

Call:
flexmix(formula = x ~ switch | consumer, k = 1)

       prior size post>0 ratio
Comp.1     1  100    100     1

’log Lik.’ -206.5132 (df=11)
AIC: 435.0264   BIC: 463.6833

> parameters(res1)
$coef
(Intercept)     switch2     switch3     switch4     switch5     switch6
        7.8         0.3        -5.6        -4.5        -0.5        -0.5
    switch7     switch8     switch9    switch10
       -4.3        -2.4        -0.8        -4.7

$sigma
[1] 2.006102
```

```
# model with 2 latent classes
> res2=flexmix(x~switch|consumer,k=2)
> summary(res2)

Call:
flexmix(formula = x ~ switch | consumer, k = 2)

       prior size post>0 ratio
Comp.1 0.301   30     80 0.375
Comp.2 0.699   70    100 0.700

’log Lik.’ -196.5404 (df=23)
AIC: 439.0809   BIC: 498.9998

# model with 3 latent classes
> res3=flexmix(x~switch|consumer,k=3)
> summary(res3)

Call:
flexmix(formula = x ~ switch | consumer, k = 3)

       prior size post>0 ratio
Comp.1 0.399   40     80 0.500
Comp.2 0.401   40     70 0.571
Comp.3 0.200   20     20 1.000

’log Lik.’ -167.5131 (df=35)
AIC: 405.0263   BIC: 496.2072


# AIC does not increase with 3 classes
# therefor 2 classes are asumed (res2)

# info on each class
> parameters(res2,component=1)
$coef
(Intercept)     switch2     switch3     switch4     switch5     switch6
  6.0026862   1.0056101  -3.6616921  -1.3610496   0.3529849   1.0072038
    switch7     switch8     switch9    switch10
 -2.3399164   2.3180905   2.6458269  -4.9939205

$sigma
[1] 1.660916

> parameters(res2,component=2)
$coef
 (Intercept)      switch2      switch3      switch4      switch5
 8.575795002 -0.004570537 -6.436653893 -5.854900878 -0.868183600
     switch6      switch7      switch8      switch9     switch10
-1.150571545 -5.146053205 -4.436523113 -2.287361487 -4.573131763

$sigma
[1] 1.732276

```

```
# theta for each class assigned to p1 and p2
> p1=parameters(res2,component=1)$coef
> p2=parameters(res2,component=2)$coef

> p1
(Intercept)     switch2     switch3     switch4     switch5     switch6
  6.0026862   1.0056101  -3.6616921  -1.3610496   0.3529849   1.0072038
    switch7     switch8     switch9    switch10
 -2.3399164   2.3180905   2.6458269  -4.9939205
> p2
 (Intercept)      switch2      switch3      switch4      switch5
 8.575795002 -0.004570537 -6.436653893 -5.854900878 -0.868183600
     switch6      switch7      switch8      switch9     switch10
-1.150571545 -5.146053205 -4.436523113 -2.287361487 -4.573131763

# class observation for each consumer
> posterior(res2) # three persons differ from the rest...
               [,1]         [,2]
  [1,] 9.998531e-01 0.0001469120
  [2,] 9.998531e-01 0.0001469120
  [3,] 9.998531e-01 0.0001469120
  [4,] 9.998531e-01 0.0001469120
  [5,] 9.998531e-01 0.0001469120
  [6,] 9.998531e-01 0.0001469120
  [7,] 9.998531e-01 0.0001469120
  [8,] 9.998531e-01 0.0001469120
  [9,] 9.998531e-01 0.0001469120
 [10,] 9.998531e-01 0.0001469120
 [11,] 3.332089e-04 0.9996667911
 [12,] 3.332089e-04 0.9996667911
 [13,] 3.332089e-04 0.9996667911
 [14,] 3.332089e-04 0.9996667911
 [15,] 3.332089e-04 0.9996667911
 [16,] 3.332089e-04 0.9996667911
 [17,] 3.332089e-04 0.9996667911
 [18,] 3.332089e-04 0.9996667911
 [19,] 3.332089e-04 0.9996667911
 [20,] 3.332089e-04 0.9996667911
 [21,] 1.203385e-03 0.9987966149
 [22,] 1.203385e-03 0.9987966149
 [23,] 1.203385e-03 0.9987966149
 [24,] 1.203385e-03 0.9987966149
 [25,] 1.203385e-03 0.9987966149
 [26,] 1.203385e-03 0.9987966149
 [27,] 1.203385e-03 0.9987966149
 [28,] 1.203385e-03 0.9987966149
 [29,] 1.203385e-03 0.9987966149
 [30,] 1.203385e-03 0.9987966149
 [31,] 4.888132e-08 0.9999999511
 [32,] 4.888132e-08 0.9999999511
 [33,] 4.888132e-08 0.9999999511
 [34,] 4.888132e-08 0.9999999511
 [35,] 4.888132e-08 0.9999999511
 [36,] 4.888132e-08 0.9999999511
 [37,] 4.888132e-08 0.9999999511
```

```
[38,] 4.888132e-08 0.9999999511
[39,] 4.888132e-08 0.9999999511
[40,] 4.888132e-08 0.9999999511
[41,] 1.564294e-03 0.9984357058
[42,] 1.564294e-03 0.9984357058
[43,] 1.564294e-03 0.9984357058
[44,] 1.564294e-03 0.9984357058
[45,] 1.564294e-03 0.9984357058
[46,] 1.564294e-03 0.9984357058
[47,] 1.564294e-03 0.9984357058
[48,] 1.564294e-03 0.9984357058
[49,] 1.564294e-03 0.9984357058
[50,] 1.564294e-03 0.9984357058
[51,] 9.959409e-01 0.0040591087
[52,] 9.959409e-01 0.0040591087
[53,] 9.959409e-01 0.0040591087
[54,] 9.959409e-01 0.0040591087
[55,] 9.959409e-01 0.0040591087
[56,] 9.959409e-01 0.0040591087
[57,] 9.959409e-01 0.0040591087
[58,] 9.959409e-01 0.0040591087
[59,] 9.959409e-01 0.0040591087
[60,] 9.959409e-01 0.0040591087
[61,] 9.990717e-01 0.0009282781
[62,] 9.990717e-01 0.0009282781
[63,] 9.990717e-01 0.0009282781
[64,] 9.990717e-01 0.0009282781
[65,] 9.990717e-01 0.0009282781
[66,] 9.990717e-01 0.0009282781
[67,] 9.990717e-01 0.0009282781
[68,] 9.990717e-01 0.0009282781
[69,] 9.990717e-01 0.0009282781
[70,] 9.990717e-01 0.0009282781
[71,] 8.140028e-08 0.9999999186
[72,] 8.140028e-08 0.9999999186
[73,] 8.140028e-08 0.9999999186
[74,] 8.140028e-08 0.9999999186
[75,] 8.140028e-08 0.9999999186
[76,] 8.140028e-08 0.9999999186
[77,] 8.140028e-08 0.9999999186
[78,] 8.140028e-08 0.9999999186
[79,] 8.140028e-08 0.9999999186
[80,] 8.140028e-08 0.9999999186
[81,] 1.270020e-03 0.9987299803
[82,] 1.270020e-03 0.9987299803
[83,] 1.270020e-03 0.9987299803
[84,] 1.270020e-03 0.9987299803
[85,] 1.270020e-03 0.9987299803
[86,] 1.270020e-03 0.9987299803
[87,] 1.270020e-03 0.9987299803
[88,] 1.270020e-03 0.9987299803
[89,] 1.270020e-03 0.9987299803
[90,] 1.270020e-03 0.9987299803
[91,] 1.573586e-02 0.9842641441
[92,] 1.573586e-02 0.9842641441
```

```
 [93,] 1.573586e-02 0.9842641441
 [94,] 1.573586e-02 0.9842641441
 [95,] 1.573586e-02 0.9842641441
 [96,] 1.573586e-02 0.9842641441
 [97,] 1.573586e-02 0.9842641441
 [98,] 1.573586e-02 0.9842641441
 [99,] 1.573586e-02 0.9842641441
[100,] 1.573586e-02 0.9842641441

mrank=ave(x,switch)[1:10]


# ranks for class 1 (without consumer 1,7)
> x1=x[!(consumer==7|consumer==1)]
# ranks for class 2 (consumer 1, 7)
> x2=x[(consumer==7|consumer==1)]

# switchnumbers for each class
> switch1=switch[!(consumer==7|consumer==1)]
> switch2=switch[(consumer==7|consumer==1)]

# mean of ranks of each class
> m1rank=ave(x1,switch1)[1:10]
> m2rank=ave(x2,switch2)[1:10]

> mrank
 [1] 7.8 8.1 2.2 3.3 7.3 7.3 3.5 5.4 7.0 3.1
> m1rank
 [1] 8.375 8.750 2.125 3.500 7.375 7.000 3.375 4.625 6.250 3.625
> m2rank
 [1]  5.5  5.5  2.5  2.5  7.0  8.5  4.0  8.5 10.0  1.0
> dev.off()
Error in dev.off() : cannot shut down device 1 (the null device)


# ranks for class 1 (without consumer 1, 6 and 7)
> x1=x[!(consumer==7|consumer==6|consumer==1)]
# ranks for class 2 (consumer 1, 6 and 7)
> x2=x[(consumer==7|consumer==6|consumer==1)]

> x1
 [1]  8 10  3  2  7  4  1  5  9  6  9  7  2  1  6 10  3  5  8  4 10  9  2  1  7
[26]  8  5  4  3  6  8 10  1  6  9  7  3  5  4  2 10  9  2  3  8  7  6  1  5  4
[51]  9  6  1  5  7  8  3  2 10  4  6  9  4  1 10  8  3  7  5  2
> x2
 [1]  6  5  2  3  7  8  4  9 10  1  7 10  2  9  5  4  3  8  6  1  5  6  3  2  7
[26]  9  4  8 10  1

# switchnumbers for each class
> switch1=switch[!(consumer==7|consumer==6|consumer==1)]
> switch2=switch[(consumer==7|consumer==6|consumer==1)]

> switch1
 [1] 1  2  3  4  5  6  7  8  9  10 1  2  3  4  5  6  7  8  9  10 1  2  3  4  5
[26] 6  7  8  9  10 1  2  3  4  5  6  7  8  9  10 1  2  3  4  5  6  7  8  9  10
```

```
[51] 1  2  3  4  5  6  7  8  9  10 1  2  3  4  5  6  7  8  9  10
Levels: 1 2 3 4 5 6 7 8 9 10
> switch2
 [1] 1  2  3  4  5  6  7  8  9  10 1  2  3  4  5  6  7  8  9  10 1  2  3  4  5
[26] 6  7  8  9  10
Levels: 1 2 3 4 5 6 7 8 9 10

# mean of ranks of each class
> m1rank=ave(x1,switch1)[1:10]
> m2rank=ave(x2,switch2)[1:10]

> m1rank
 [1] 8.571429 8.571429 2.142857 2.714286 7.714286 7.428571 3.428571 4.142857
 [9] 6.285714 4.000000
> m2rank
 [1] 6.000000 7.000000 2.333333 4.666667 6.333333 7.000000 3.666667 8.333333
 [9] 8.666667 1.000000
```

latentanalysis.m

```
clear all
close all
%for title, label, legend
set(0,'defaultaxesfontsize',13, 'defaultaxesfontweight','bold');
%for line and marker
set(0,'DefaultLineLineWidth',2,'DefaultLineMarkerSize',10 );
%text ??
set(0,'defaulttextfontsize',13, 'defaulttextfontweight','bold');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%this script contain info about the Latent Class Analysis of Q7.

%how many latent classes?
%AIC for 1, 2 and 3 classes.. from R code
aic = [435.0264, 439.0809, 405.0263];


%the posterior probabilities for each consumer indicating,
% that consumer 1, 6 and 7 differ from the rest.

%load posterior.m, to get posterior data
posterior

post = post(10*(1:10),:);

figure
plot([2,3,4,5,8,9,10],post([2,3,4,5,8,9,10]),'or'); hold on;
plot([1,6,7],post([1,6,7]),'ob');
xlabel('consumer')
ylabel('percent')
legend('consumer 2,3,4,5,8,9,10','consumer 1,6,7','Location', 'NorthEast')
set(gca,'xtick',linspace(1,10,10),'Box','off', 'xlim', [0,11], ...
     'ylim',[-1,2]);

%rank mean class 1 (consumer 2,3,4,5,8,9,10) and class 2 (consumer 1,6,7)
mrank = [7.8 8.1 2.2 3.3 7.3 7.3 3.5 5.4 7.0 3.1];
m1rank =[8.571429 8.571429 2.142857 2.714286 7.714286
```

```
         7.428571 3.428571 4.142857 6.285714 4.000000];
m2rank =[6.000000 7.000000 2.333333 4.666667 6.333333
         7.000000 3.666667 8.333333 8.666667 1.000000];

[mrankpermut,permut] = sort(mrank,’ascend’)
m1rankpermut = m1rank(permut);
m2rankpermut = m2rank(permut);

figure %latentmean.eps
plot(mrankpermut, ’ok-’); hold on;
plot(m1rankpermut, ’r-’);
plot(m2rankpermut,’b-’);
xlabel(’switch’)
ylabel(’rank mean’)
legend(’mean’,’class 1 mean’, ’class 2 mean’, ’Location’, ’SouthEast’)
set(gca,’xtick’,linspace(1,10,10),’XTickLabel’,permut,’Box’,’off’,
        ’ytick’, linspace(1,10,10), ’xlim’, [0,11], ’ylim’,[-0.5,11.5]);


%theta for class 1 and 2
theta1_ste1=[-1.85448    0.30531
    -1.88224    0.30705
     0.84368    0.28822
     0.53934    0.27149
    -1.49968    0.28604
    -1.35833    0.27967
     0.23201    0.26097
     -0.08213    0.25566
     -0.89304    0.26348
     0          0];

theta2_ste2=[-7.355  ,  262.462;
      -7.424  ,  451.915;
      -7.789  ,  451.915;
      -5.496  ,  451.915;
      -6.915  ,  451.915;
      -7.531  ,  451.915;
      -6.382  ,  451.915;
      -8.303  ,  451.915;
      -8.432  ,  451.915;
      0,0];

theta =[-1.6657386, -1.7972956 , 0.4268400, -0.1019803, -1.4933180,
            -1.4832802, -0.1461087, -0.8337355, -1.3615162,0] ;
theta1 = theta1_ste1(:,1)’;
theta2 = theta2_ste2(:,1)’;

thetapermut = -theta(permut);
theta1permut=-theta1(permut);
theta2permut=-theta2(permut);

figure %latenttheta.eps
plot(thetapermut, ’ok-’); hold on;
plot(theta1permut, ’r-’);
plot(theta2permut,’b-’);
```

```
xlabel(’switch’)
ylabel(’-\theta’)
legend(’-\theta’,’-\theta, class 1’, ’-\theta, class 2’, ...
    ’Location’, ’SouthEast’)
set(gca,’xtick’,linspace(1,10,10),’XTickLabel’,permut,’Box’,’off’, ...
    ’xlim’, [0,11]);
```

binprobitclasses.R

```
# estimate theta for each class.

ls()
search()

#Load data into data frame:
BOQ <-read.table(’RQ7class1.txt’,header=T,sep=’     ’)
#n = rep(10,45) # if RQ4 n=rep(20,45) if RQ6 n=rep(18,45)
n=rep(7,45) #class1
#n=rep(3,45) #class2

#Add the data to the search path:
attach(BOQ)

# GLM (Bradley-Terry -probit) for consumer 2,3,4,5,8,9,10
model <- glm(cbind(Y,n-Y) ~ X1+X2+X3+X4+X5+X6+X7+X8+X9+X10-1,
             family=binomial(link=probit), data=BOQ)
coef(model)
summary(model)


# Clear the work space and search path:
rm(list=ls(all=TRUE))
detach(BOQ)


###### CLASS 1 ##################
> coef(model)
         X1          X2          X3          X4          X5          X6
-1.85447712 -1.88224084  0.84368479  0.53934327 -1.49968203 -1.35832721
         X7          X8          X9         X10
 0.23201411 -0.08213204 -0.89303875          NA
> summary(model)

Call:
glm(formula = cbind(Y, n - Y) ~ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3088  -0.6006  -0.2211   0.3666   1.1211

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1  -1.85448    0.30531  -6.074 1.25e-09 ***
```

```
X2  -1.88224    0.30705  -6.130 8.78e-10 ***
X3   0.84368    0.28822   2.927  0.00342 **
X4   0.53934    0.27149   1.987  0.04697 *
X5  -1.49968    0.28604  -5.243 1.58e-07 ***
X6  -1.35833    0.27967  -4.857 1.19e-06 ***
X7   0.23201    0.26097   0.889  0.37398
X8  -0.08213    0.25566  -0.321  0.74802
X9  -0.89304    0.26348  -3.389  0.00070 ***
X10       NA         NA      NA       NA
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 217.126  on 45  degrees of freedom
Residual deviance:  17.520  on 36  degrees of freedom
AIC: 95.836

Number of Fisher Scoring iterations: 6

###### CLASS 2 ##################################
> coef(model)
       X1        X2        X3        X4        X5        X6        X7        X8
-7.423629 -7.797988 -5.495509 -6.915105 -7.530833 -7.779726 -6.381677 -8.302855
       X9       X10
-8.431670        NA
> summary(model)

Call:
glm(formula = cbind(Y, n - Y) ~ X1 + X2 + X3 + X4 + X5 + X6 +
    X7 + X8 + X9 + X10 - 1, family = binomial(link = probit),
    data = BOQ)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-1.387e+00  -2.692e-01  -1.366e-07   3.561e-01   1.336e+00

Coefficients: (1 not defined because of singularities)
    Estimate Std. Error z value Pr(>|z|)
X1    -7.424    451.915  -0.016    0.987
X2    -7.798    451.915  -0.017    0.986
X3    -5.496    451.915  -0.012    0.990
X4    -6.915    451.915  -0.015    0.988
X5    -7.531    451.915  -0.017    0.987
X6    -7.780    451.915  -0.017    0.986
X7    -6.382    451.915  -0.014    0.989
X8    -8.303    451.915  -0.018    0.985
X9    -8.432    451.915  -0.019    0.985
X10       NA         NA      NA       NA

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 118.406  on 45  degrees of freedom
Residual deviance:  21.319  on 36  degrees of freedom
AIC: 68.513
```

```

Number of Fisher Scoring iterations: 18
```

Bibliography

[1] Alan Agresti. *Categorical Data Analysis*. Whiley, 2002. Second Edition.

[2] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] Knut Conradsen. *En introduktion til statestik, Bind 2*. IMM, DTU, 2003.

[4] Philippe Courcoux and Michael Semenou. Preference data analysis using a paired comparison model. *Food Quality and Preference*, 8, No.5/6:353–358, 1997.

[5] Douglas E Critchlow and Michael A Fligner. Paired comparison, triple comparison, and ranking experiments as generalized linear models, and their implementation on glim. *Psychometrika*, 56, No. 3:517–533, 1991.

[6] Philippe Courcoux E Vigneau and Michal Semenou. Analysis of ranked preference data using latent class models. *Food Quality and Preference*, 10:201–207, 1999.

[7] Michael A Fligner and Joseph S Verducci. Distance based ranking models. *Journal of the Royal Statistical Society*, 48, No. 3:359–369, 1986.

[8] Michael A. Fligner and Joseph S. Verducci. Multistage ranking models. *Journal of Amarican Statistical Association*, 83, No. 403:892–901, 1988.

[9] Michael A. Fligner and Joseph S. Verducci. *Ranking Models with Item Covariates*. Springer-Verlag, 1992. Foreword.

[10] Inge Henningsen. *Statistik*. Institut for Matematisk Statistik, KU, 1996. 6. udgave.

[11] Friedrich Leisch. Flexmix: A general framework for finite mixture models and latent class regression in r. *Journal of Statistical Software*, 11, Issur 8, 2004.

[12] Kaj Madsen and Hans Bruun Nielsen. Supplementary notes for 02611 optimization and data fitting. Technical report, IMM, DTU, 2004.

[13] Kaj Madsen, Hans Bruun Nielsen, and O. Tingleff. Methods for non-linear least squares problems. Technical report, IMM,DTU, 2004.

[14] Ditte Hvas Mortensen and Søren Bech. The relative importance of visuel, auditory and haptic information for the user's experience of mechanical switches. Technical report, Bang & Olufsen a/s, 2006.

[15] S Nadarajah. Sums, products, and ratios for the bivariate gumbel distribution. *Elsevier*, 42:499–518, 2005.

[16] Ulf Olsson. *Generalized Linear Models*. Studentlitteratur, 2002. An Applied Approach.

[17] Louis Leon Thurstone. Attitudes can be measured. *The Amarical Journal of Socialogi*, 33, No. 4:529–554, 1928.

[18] Poul Thyregod. Noter til specialkusus i statestik, kap. 4. Technical report, IMM,DTU, 2005.

[19] Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Statical Learning*. Springer, 2001.

[20] William M.K. Trochim. www.socialresearchmethods.net/kb/scaling.php. Technical report, Cornell University, 2006.

[21] ukendt. kapitel 9, tærskelmodeller, udleveret af pbb. Technical report, IMM,DTU, 2007.