



## **BTLasso: A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models**

**Gunther Schaubeger**  
Technical University of Munich  
LMU Munich

**Gerhard Tutz**  
LMU Munich

---

### **Abstract**

In paired comparison models, the inclusion of covariates is a tool to account for the heterogeneity of preferences and to investigate which characteristics determine the preferences. Although methods for the selection of variables have been proposed no coherent framework that combines all possible types of covariates is available. There are three different types of covariates that can occur in paired comparisons, the covariates can either vary over the subjects, the objects or both the subjects and the objects of the paired comparisons. This paper gives an overview over all possible types of covariates in paired comparisons and introduces a general framework to include covariate effects into Bradley-Terry models. For each type of covariate, appropriate penalty terms that allow for sparser models and, therefore, easier interpretation are proposed. The whole framework is implemented in the R package **BTLasso**. The main functionality and the visualization tools of the package are introduced and illustrated by real data sets.

*Keywords:* Bradley-Terry, paired comparison, penalization, variable selection, **BTLasso**.

---

## **1. Introduction**

Paired comparisons have a long tradition in statistics with the roots dating back to [Zermelo \(1929\)](#). In many applications the objective is to find the underlying preference scale by presenting objects or items in pairs. The method has been used in various areas, for example, in psychology, to measure the intensity or attractiveness of stimuli, in marketing, to evaluate the attractiveness of brands, in social sciences, to investigate value orientation, e.g., [Francis, Dittrich, Hatzinger, and Penn \(2002\)](#). Paired comparisons are also found in sports whenever two players or teams compete. Then the non-observable scale to be found refers to the strengths of the competitors.

The most widely used model for paired comparison data is the Bradley-Terry model. Originally proposed by [Bradley and Terry \(1952\)](#) it is also referred to as the Bradley-Terry-Luce (BTL) model indicating the strong connection to Luce's choice axiom formulated in [Luce \(1959\)](#). Luce's choice axiom states that the decision between two objects is not influenced by other objects, which is also known as the independence from irrelevant alternatives. Over the years various extensions have been proposed allowing for dependencies among responses, time dependence or simultaneous ranking with respect to more than one attribute. Overviews were given by [Bradley \(1976\)](#), [David \(1988\)](#) and more recently by [Cattelan \(2012\)](#).

The main problem when applying the Bradley-Terry model is that it implies severe restrictions and assumptions. In particular it assumes that there is a one-dimensional latent scale that represents the dominance, strength or attractiveness of objects. In experiments with several persons or, more generally, several subjects, the BTL model is typically fitted under the assumption that the strengths of the objects are equal for all subjects. Since the heterogeneity across subjects is ignored the model often yields a bad fit. The assumption of a latent scale becomes much weaker if one assumes that the strengths of objects depend on covariates. The covariates can be subject-specific or object-specific. The former refers to the characteristics of the subjects who choose between alternatives/objects, the latter refers to the characteristics of the objects which are compared. Also subject-object-specific covariates, that is, characteristics that vary both over subjects and over objects, are possible. Explicit modeling of heterogeneity is rare, and usually restricted to few covariates, see [Turner and Firth \(2012\)](#) or [Francis, Dittrich, and Hatzinger \(2010\)](#). Software packages for the analysis of paired comparisons that are available as R packages ([R Core Team 2019](#)) are the package **BradleyTerry2** ([Turner and Firth 2012](#)) and **prefmod** ([Hatzinger and Dittrich 2012](#)). Both allow for the integration of parametric effects for object-specific and subject-specific covariates into paired comparison models. To make models with object-specific covariates identifiable, package **BradleyTerry2** uses random effects for the main object parameters. In package **prefmod**, the object parameters are excluded. However, the packages do not contain built-in procedures for the selection and clustering of covariate effects and do not use regularized estimation. Also, none of the packages allows for object-specific order effects. More recently, methods that are able to handle a larger number of explanatory variables have been proposed. [Casalicchio, Tutz, and Schauburger \(2015\)](#) presented a boosting approach and developed the corresponding R package **ordBTL** ([Casalicchio 2014](#)). The approach is restricted to subject-specific covariates. Also, the boosting approach (in contrast to penalization as proposed in this work) is not designed to cluster effects. [Strobl, Wickelmaier, and Zeileis \(2011\)](#) used recursive partitioning techniques and implemented partitioned paired comparison models in the R package **psychotree** ([Zeileis, Strobl, Wickelmaier, Komboz, and Kopf 2018](#)). This approach generates sub-spaces of the predictor space where identical Bradley-Terry models hold, the models differ between the sub-spaces. The results can be visualized by trees which provide a very intuitive and user-friendly interpretation when the main focus is to find sub-groups of subjects with equal preferences. The approach is restricted to subject-specific covariates.

The objective of the present paper is to develop a common framework for the inclusion and selection of covariates in BTL models based on penalization techniques. The inclusion of covariates serves two purposes. It accounts for heterogeneity in the population yielding more realistic models and it allows to investigate which variables determine the strengths of objects. The proposed methods are implemented in the R package **BTLasso** ([Schauburger 2019](#)) available from the Comprehensive R Archive Network (CRAN) at <https://>

[//CRAN.R-project.org/package=BTLLasso](http://CRAN.R-project.org/package=BTLLasso). In the following also the use of the package is described and illustrated.

## 2. The Bradley-Terry model

In the following the simple binary Bradley-Terry model and extended versions that allow for ordinal responses are briefly sketched.

### 2.1. The basic model

Assuming a set of objects  $\{a_1, \dots, a_m\}$ , in its most simple form the (binary) Bradley-Terry model is given by

$$P(a_r > a_s) = P(Y_{(r,s)} = 1) = \frac{\exp(\gamma_r - \gamma_s)}{1 + \exp(\gamma_r - \gamma_s)}. \quad (1)$$

The response of the model represents the probability that a certain object  $a_r$  dominates (or is preferred over) another object  $a_s$ ,  $a_r > a_s$ . The response is formalized in the random variable  $Y_{(r,s)}$  which is defined to be  $Y_{(r,s)} = 1$  if  $a_r$  is preferred over  $a_s$  and  $Y_{(r,s)} = 0$  otherwise. The parameters  $\gamma_r$ ,  $r = 1, \dots, m$ , represent the attractiveness or strength of the respective objects. For identifiability, a restriction on the parameters is needed, for example  $\sum_{r=1}^m \gamma_r = 0$  or  $\gamma_m = 0$ .

### 2.2. Bradley-Terry models with ordered response categories

In many applications the dominance of one of the objects is quite naturally observed on an ordered scale. For example, in sport competitions one often has to account for draws. Early extensions of the BTL model include at least the possibility of ties, see [Rao and Kupper \(1967\)](#), [Glenn and David \(1960\)](#) and [Davidson \(1970\)](#). General models for ordered responses that allow, for example, for a 5-point scale (much better, slightly better, equal, slightly worse, much worse) were proposed by [Tutz \(1986\)](#) and [Agresti \(1992\)](#). For  $K$  response categories, the model parameterizes the cumulative probabilities

$$P(Y_{(r,s)} \leq k) = \frac{\exp(\theta_k + \gamma_r - \gamma_s)}{1 + \exp(\theta_k + \gamma_r - \gamma_s)} \quad (2)$$

with  $k = 1, \dots, K$  denoting the possible response categories. The parameters  $\theta_k$  represent so-called threshold parameters for the single response categories, they determine the preference for specific categories. In particular,  $Y_{(r,s)} = 1$  represents the maximal preference for object  $a_r$  over  $a_s$  and  $Y_{(r,s)} = K$  represents the maximal preference for object  $a_s$  over  $a_r$ . For ordinal paired comparisons it can be assumed that the response categories have a symmetric interpretation so that  $P(Y_{(r,s)} = k) = P(Y_{(s,r)} = K - k + 1)$  holds. Therefore, the threshold parameters should be restricted. One postulates  $\theta_k = -\theta_{K-k}$  and, if  $K$  is even,  $\theta_{K/2} = 0$  to guarantee symmetric probabilities. The threshold for the last category is fixed to  $\theta_K = \infty$  so that  $P(Y_{(r,s)} \leq K) = 1$  holds. The probability for a single response category can be derived from the difference between two adjacent categories,  $P(Y_{(r,s)} = k) = P(Y_{(r,s)} \leq k) - P(Y_{(r,s)} \leq k - 1)$ . To guarantee non-negative probabilities for the single response categories one restricts  $\theta_1 \leq \theta_2 \leq \dots \leq \theta_K$ . The ordinal Bradley-Terry model is a cumulative logit model and can be estimated using methods from this general framework.

### 2.3. Bradley-Terry models with order effects

In some specific paired comparisons it can be decisive in which order the competing objects are presented. Typical examples are sports events, for example, matches in the German Bundesliga which are considered as an exemplary data set in this article. In this application, the first team of the ordered tuple  $(a_r, a_s)$  is the team playing at its home ground where it might have a home advantage over its opponent. Therefore, the assumption that the response categories are symmetric does not hold anymore and the model needs to be adapted accordingly. Extending the basic models by an additional parameter  $\delta$  yields the binary Bradley-Terry model with an order effect

$$P(Y_{(r,s)} = 1) = \frac{\exp(\delta + \gamma_r - \gamma_s)}{1 + \exp(\delta + \gamma_r - \gamma_s)}$$

and the ordinal model with an order effect

$$P(Y_{(r,s)} \leq k) = \frac{\exp(\delta + \theta_k + \gamma_r - \gamma_s)}{1 + \exp(\delta + \theta_k + \gamma_r - \gamma_s)}. \quad (3)$$

Here,  $\delta$  denotes the order effect which is simply incorporated into the design matrix by an additional intercept column. If  $\delta > 0$ , it increases the probability of the first object  $a_r$  to win the comparison or, in the case of an ordinal response, to achieve a superior result. Given the order effect, the symmetry assumption for the response categories still holds.

## 3. The general Bradley-Terry model with explanatories

In many applications covariates that characterize the subjects and/or the objects are available. These can be used to model the heterogeneity in the population and to investigate which variables determine the choice between alternatives. We first describe two applications that will be used later, the respective data sets can be found in the package **BTLLasso**.

### 3.1. Exemplary data sets

#### *Election data*

The German Longitudinal Election Study (GLES) is a long-term study of the German electoral process, see [Rattinger, Roßteutscher, Schmitt-Beck, Weßels, and Wolf \(2014\)](#). It collects pre- and post-election data for several federal elections. The specific data we are using here originate from the pre-election survey for the German federal election in 2013. In this specific part of the study, 2003 persons were asked to rate the most important parties (CDU/CSU, SPD, Greens, Left Party, FDP) for the upcoming federal election on a scale from  $-5$  to  $+5$ . The rating scales  $Z_{ir}$  reflect the general opinions of participant  $i$  on party  $a_r$  with  $+5$  representing a very positive and  $-5$  representing a very negative opinion. The original scales were transformed into paired comparisons by building all pairwise differences of the scores of all five parties per person. The transformation of rating scales to ordered paired comparison data was proposed by [Dittrich, Francis, Hatzinger, and Katzenbeisser \(2007\)](#). They also describe in detail the advantages of the transformation for the analysis of rating scales. Using this procedure one ends up with ordered paired comparisons with values between  $-10$  and  $10$ .

The response was narrowed down to an ordered response with five categories. The data now represent paired comparisons between all parties measured on an ordered five-point scale:

$$\begin{aligned} Z_{ir} - Z_{is} \in \{6, \dots, 10\} &\mapsto Y_{i(r,s)} = 1 : \text{“Strong preference of party } a_r \text{ over party } a_s\text{.”} \\ Z_{ir} - Z_{is} \in \{1, \dots, 5\} &\mapsto Y_{i(r,s)} = 2 : \text{“Weak preference of party } a_r \text{ over party } a_s\text{.”} \\ Z_{ir} - Z_{is} = 0 &\mapsto Y_{i(r,s)} = 3 : \text{“Equal opinions on parties } a_r \text{ and } a_s\text{.”} \\ Z_{ir} - Z_{is} \in \{-5, \dots, -1\} &\mapsto Y_{i(r,s)} = 4 : \text{“Weak preference of party } a_s \text{ over party } a_r\text{.”} \\ Z_{ir} - Z_{is} \in \{-10, \dots, -6\} &\mapsto Y_{i(r,s)} = 5 : \text{“Strong preference of party } a_s \text{ over party } a_r\text{.”} \end{aligned}$$

There is a crucial difference compared to the simple paired comparison models from Section 2. In addition to the compared objects  $a_r$  and  $a_s$ , the response  $Y_{i(r,s)}$  now also depends on the subject  $i$  that executes the comparison between both objects.

The transformation of rating scale data to paired comparisons is not without problems. First and foremost, when calculating the differences between two rating scales  $Z_{ir} - Z_{is}$  one assumes more than an ordinal scale level. Although this is frequently done in the literature it is certainly not optimal. In the present application the effect of building differences is mitigated by using a strongly coarsened response scale. The 21-point scale produced by differences in Likert ratings is narrowed down to a 5-point scale, for which the ordinal scale level is assumed. The less categories one uses for the resulting paired comparisons, the weaker are the assumptions with respect to the scale level of the rating scales. A three-point scale built from just distinguishing between positive, negative and zero differences would fully respect the ordinal nature of the scales. A scale of this type was used by [Dittrich \*et al.\* \(2007\)](#). Second, paired comparisons from rating scales are strictly transitive. This entails that for a single subject certain patterns of paired comparisons are not observed. [Dittrich \*et al.\* \(2007\)](#) explicitly account for the reduced number of possible patterns that can be derived from the ratings. Lastly, the use of rating scales could contribute to make paired comparisons dependent. Dependence of observations in paired comparison data stemming from various individuals is a general problem. [Dittrich, Hatzinger, and Katzenbeisser \(2002\)](#) use interaction terms that account for the dependencies of pairs of paired comparisons when both refer to a joint object. In [Dittrich \*et al.\* \(2007\)](#) the interaction terms are included for paired comparison data obtained from rating scales. The current approach avoids this rather parameter intensive parameterization and tries to account for dependencies by including various subject-specific covariates.

Two different types of covariates are available in the GLES data set, namely subject-specific covariates and subject-object-specific covariates. While the first type of covariates only characterizes the participants of the study, the second type of covariates characterizes both the participants and the parties. A more detailed distinction between the possible types of covariates in paired comparisons follows in Section 3.2. For sake of simplicity, we restrict the analysis to include only the following three covariates (out of eight covariates in the data set GLES in the package) that characterize the participants of the study:

- **Age:** age of participant in years.
- **Gender:** female (1); male (0).
- **Abitur:** school leaving certificate (1: Abitur/A levels; 0: else).

The main question regarding these variables will be, if and how they influence the preference for single parties. In addition to these person-specific covariates also three covariates are available that vary both over the persons and over the parties. These variables measure the (self-perceived) distances between participants and the single parties with respect to various political topics. These so-called political dimensions refer to the socio-economic dimension and the topics of immigration and climate change. The participants are supposed to report both their self-perception and their perception of the single parties toward a topic on a Likert scale with 11 ordered levels. For example, for the topic of climate change the Likert scale offers a graded response for the following statements:

Level 1: Fight against climate change should take precedence, even if it impairs economic growth.

Level 11: Economic growth should take precedence, even if it impairs the fight against climate change.

The absolute difference between these perceptions can be seen as the (self-perceived) absolute distance between the person (subject) and the party (object) with respect to a political topic. Therefore, the variables `SocEc`, `Immigration` and `Climate` are available separately for all five parties.

### *Football data*

The second data set we consider are data from the season 2015/2016 of the German Bundesliga. The German Bundesliga is played as a double round robin between 18 teams. Table 1 shows the final table of the season 2015/16 together with the abbreviations of the teams used in the application later. As in all three previous seasons, Bayern München won the championship. VfB Stuttgart and Hannover 96 were relegated to the second division.

In the data set, one match is treated as a paired comparison of both teams with respect to their playing abilities. In total, a Bundesliga season has 34 match days with 9 matches per match day. Therefore, in total 306 matches are observed. The response variable  $Y_{i(r,s)}$  represents the outcome of a match between team  $a_r$  (as the home team) and team  $a_s$  on match day  $i$ . We use a 5-point scale defined by

$$Y_{i(r,s)} = \begin{cases} 1 & \text{if team } a_r \text{ wins by at least 2 goals difference.} \\ 2 & \text{if team } a_r \text{ wins by 1 goal difference.} \\ 3 & \text{if the match ends with a draw.} \\ 4 & \text{if team } a_s \text{ wins by 1 goal difference.} \\ 5 & \text{if team } a_s \text{ wins by at least 2 goals difference.} \end{cases}$$

In addition to the match results encoded by ordinal paired comparisons several covariates are given. In particular, specific performance measurements of the single teams in single matches are available. For example, for every team it is known which distance the team had run in a certain match or the percentage of ball possession per team. In the German Bundesliga, the data supplier opta (<http://www.optasports.com/>) provides the data. The data we use are freely available from the website of the German football magazine kicker (<http://www.kicker.de/>). Again, only two covariates (out of seven covariates in the data set `Buli1516` in the package) are considered (per team and per match):

Position		Team	Goals For	Goals Against	Points
1		BAY Bayern München	80	17	88
2		DOR Borussia Dortmund	82	34	78
3		LEV Bayer 04 Leverkusen	56	40	60
4		MGB Bor. Mönchengladbach	67	50	55
5		S04 FC Schalke 04	51	49	52
6		MAI 1. FSV Mainz 05	46	42	50
7		BER Hertha BSC	42	42	50
8		WOB VfL Wolfsburg	47	49	45
9		KOE 1. FC Köln	38	42	43
10		HSV Hamburger SV	40	46	41
11		ING FC Ingolstadt 04	33	42	40
12		AUG FC Augsburg	42	52	38
13		BRE Werder Bremen	50	65	38
14		DAR SV Darmstadt 98	38	53	38
15		HOF TSG Hoffenheim	39	54	37
16		FRA Eintracht Frankfurt	34	52	36
17		STU VfB Stuttgart	50	75	33
18		HAN Hannover 96	31	62	25

Table 1: Final table of the German Bundesliga in the season 2015/2016.

- **Distance**: total amount of km run.
- **BallPossession**: percentage of ball possession.

The research question here is to find out whether **Distance** and/or **BallPossession** are decisive variables for the strength of a team and if the effects differ across teams. Beside these two variables, also the average market values of the players of the single teams are known. The market values are constant across the whole season and, in contrast to the previous variables, are not specific to single match days but only to single teams. Therefore, distance and ball possession are subject-object-specific covariates while market values are object-specific.

### 3.2. Model specification

In general, when modeling paired comparison data three different types of covariates have to be distinguished. All three types of covariates occur in the exemplary data presented in Section 3.1 and will be described in detail in the following. However, first a clear distinction between subjects and objects in paired comparisons has to be made. In our notation, objects refer to the units that are compared in a paired comparison experiment. For example, in the GLES data from Section 3.1, the parties are the objects which vary in their attractiveness to voters. In football matches (as considered in the Bundesliga data from Section 3.1), the teams are the objects that are compared to measure their playing abilities. In contrast, the subjects in paired comparisons refer to the units that actively make a decision concerning the dominance of objects in a paired comparisons. In the election example the voters (or rather the participants of the study) represent the subjects. In football, the definition of subjects

is less obvious but also here subjects can be identified. We consider the day of the match or the match itself, which can be characterized by covariates, as the subject of the paired comparison.

In the following, we distinguish between subject-specific covariates  $\mathbf{x}_i$ , object-specific covariates  $\mathbf{z}_r$  and subject-object-specific covariates  $\mathbf{z}_{ir}$ . In the general framework proposed here, those covariates can be included (simultaneously) into Bradley-Terry models. For that purpose we consider the (ordinal) Bradley-Terry model in a generalized form. Let  $Y_{i(r,s)} \in 1, \dots, K$  denote the response of subject  $i$  if the objects  $(a_r, a_s)$  are presented. The ordering in the tuple  $(a_r, a_s)$  is not arbitrary, it may represent the ordering of presentation ( $a_r$  first,  $a_s$  second) or the location of the meeting of teams ( $a_r$  at home,  $a_s$  away). Then the model has the form

$$P(Y_{i(r,s)} \leq k) = \frac{\exp(\eta_{i(rs)k})}{1 + \exp(\eta_{i(rs)k})} = \frac{\exp(\delta + \theta_k + \gamma_{ir} - \gamma_{is})}{1 + \exp(\delta + \theta_k + \gamma_{ir} - \gamma_{is})}$$

or equivalently

$$\log \left( \frac{P(Y_{i(r,s)} \leq k)}{P(Y_{i(r,s)} > k)} \right) = \delta + \theta_k + \gamma_{ir} - \gamma_{is}, \quad k = 1, \dots, K - 1. \quad (4)$$

In contrast to model (3), the global object parameters  $\gamma_r$  are replaced by subject-specific parameters  $\gamma_{ir}$  for subjects  $i$ ,  $i = 1, \dots, n$  and objects  $a_r$ ,  $r = 1, \dots, m$ . The extension allows to model heterogeneity of subjects by including object-specific and subject-(object-)specific covariates. Generally, for all possible covariates we assume a linear dependence structure with the linear predictors of the model given by

$$\eta_{i(rs)k} = \delta + \theta_k + \gamma_{ir} - \gamma_{is},$$

but with the strength parameters  $\gamma_{ir}$  and  $\gamma_{is}$  depending on explanatory variables.

### *Subject-specific covariates*

Subject-specific covariates characterize the subjects that perform the comparisons between the objects. In the case of the election data, the participants of the study represent the subjects. Covariates like age, gender or unemployment status of the participants are treated as subject-specific covariates, which may modify the attractiveness of parties. This is obtained by assuming that the attractiveness of the parties (the strength parameters  $\gamma_{ir}$ ) have the form

$$\gamma_{ir} = \beta_{r0} + \mathbf{x}_i^\top \boldsymbol{\beta}_r,$$

where  $\mathbf{x}_i$  is a vector of subject-specific covariates. The inclusion of subject-specific covariates into a paired comparison model automatically requires object-specific effects  $\boldsymbol{\beta}_r$ , which vary over objects or alternatives. In the election study, for example, one wants to know if and how the attractiveness of parties depends on the gender of the voters. Consequently, for every party a distinct (object-specific) gender effect has to be specified. Accordingly, the attractiveness of object  $a_r$  depends on covariates  $\mathbf{x}_i$  of subject  $i$ . The parameter  $\beta_{rj}$  represents the increase of attractiveness of object  $r$  if the  $j$ th variable increases by one unit. To obtain identifiability of the parameters, symmetric side constraints are used, i.e., one postulates  $\sum_{r=1}^m \beta_{rj} = 0$  for all  $j$ .

The inclusion of subject-specific covariates strongly weakens the assumptions on the model. It is no longer assumed that the strengths of the objects are fixed and equal for all subjects when they choose between a pair of objects. The model explicitly accounts for heterogeneity over subjects. Subject-specific covariates in paired comparisons were considered before by [Turner and Firth \(2012\)](#), [Francis \*et al.\* \(2002\)](#) or [Francis \*et al.\* \(2010\)](#). More recently, [Casalicchio \*et al.\* \(2015\)](#) presented a boosting approach and [Schauberger and Tutz \(2017\)](#) a penalization approach based on an  $L_1$ -penalty. Both are able to include explanatory variables and select the relevant ones. An alternative approach has been proposed by [Strobl \*et al.\* \(2011\)](#). It is based on recursive partitioning techniques (also known as trees) and automatically selects the relevant variables among a potentially large set of variables.

### *Subject-object-specific covariates*

Covariates may also vary both over subjects and objects yielding subject-object-specific covariates. For example, in election studies as in the election data presented above it is frequently of interest how political issues determine the preference of specific parties. The so-called spatial election theory assumes that each party is characterized by a position in a finite-dimensional space, with each dimension corresponding to a political issue. Spatial election theory then explains the party choice of voters by a utility function that depends on the distance between the voter's own position and that of the parties within the space of policy-dimensions. For details see, for example, [Thurner and Eymann \(2000\)](#) and [Mauerer, Pössnecker, Thurner, and Tutz \(2015\)](#). In the election data, the covariates `SocEc`, `Immigration` and `Climate` measure the distance between participants and the single parties on three different political dimensions and are, therefore, considered as subject-object-specific covariates.

The effect of subject-object-specific covariates can be modeled in the same way as the effect of subject-specific covariates. With  $\mathbf{z}_{ir}$  denoting the corresponding covariate vector one uses

$$\gamma_{ir} = \beta_{r0} + \mathbf{z}_{ir}^{\top} \boldsymbol{\alpha}_r.$$

We use the notation  $\mathbf{z}_{ir}$  for the vector and  $\boldsymbol{\alpha}_r$  for the weights to distinguish the cases subject-specific and subject-object-specific covariates.

In contrast to subject-specific covariates, subject-object-specific covariates can also be modeled with global effects, which yields the simpler form

$$\gamma_{ir} = \beta_{r0} + \mathbf{z}_{ir}^{\top} \boldsymbol{\tau}.$$

The underlying assumption is that the effect is equal across all objects. That means, for example, that the distances between the positions of respondents and parties on an issue like climate change have the same relevance for all parties. With the usual constraint on the intercepts, for example,  $\sum_{r=1}^m \beta_{r0} = 0$ , parameters are identified if there is enough variation over subjects, even in the case of object-specific effects. It may be checked by investigating if the corresponding design matrix for all paired comparisons has full rank. However, due to the penalization techniques we will use for estimation the parameters will be estimable even in cases where the design matrix might not have full rank if the penalization is strong enough.

### *Object-specific covariates*

It is more difficult to include pure object-specific covariates. Object-specific covariates characterize the compared objects but are constant across the subjects. In the football data, the

market value is an example for such an object-specific variable because the market values vary across teams but are constant across match days. Similar to the procedure proposed here, [Tutz and Schaubberger \(2015\)](#) included an object-specific covariate in an analysis on the German Bundesliga.

The contribution of the object-specific covariates  $\mathbf{z}_r$  to the strength of object  $a_r$  may be specified by

$$\gamma_{ir} = \gamma_r = \beta_{r0} + \mathbf{z}_r^\top \boldsymbol{\tau},$$

where  $\boldsymbol{\tau}$  is a global parameter. Then  $\mathbf{z}_r^\top \boldsymbol{\tau}$  represents the part of the attractiveness that is due to the covariates and  $\beta_{r0}$  represents the part that is not explained by covariates. For example, in the election data if one is interested if governing makes parties attractive one can use a dummy variable  $z = 1$  for parties in the government and  $z = 0$  for parties of the opposition. Then,  $\tau$  represents the importance of being part of the government for the attractiveness of a party. In the case of object-specific covariates, no heterogeneity across subjects is modeled.

It is obvious that the parameterization leads to identifiability problems and additional side constraints are needed. Let us consider the simple example of a binary covariate, as before let  $z = 1$  indicate that the party is part of the government and  $z = 0$  indicate that the party is not part of the government. Then, the parties are nested within the factor  $z$ . Additional side constraints have to restrict the parameters within the blocks formed by  $z = 1$  and  $z = 0$ . One may use  $\sum_{j \in M_1} \beta_{j0} = 0$ ,  $\sum_{j \in M_0} \beta_{j0} = 0$ , where  $M_1$  and  $M_0$  are partitions of the parties with  $M_1$  collecting all the parties with  $z = 1$  and  $M_0$  all parties with  $z = 0$ , respectively. Identifiability issues will be considered in more detail in the following sections.

### *Intercepts*

In most models, in addition to the various covariate effects also object-specific intercepts  $\beta_{r0}$  are needed. Without covariates, the intercepts correspond to the regular strength/ability parameters from the basic Bradley-Terry models (1) or (2). Also for the intercept parameters the additional side constraint  $\sum_{r=1}^m \beta_{r0} = 0$  is necessary for identifiability.

### *Order effects*

As already mentioned in Section 2 in some circumstances the order of the competing objects can be decisive and order effects are necessary. Instead of a global order effect  $\delta$ , which is equal across all objects, also object-specific order effects  $\delta_r$  are possible. For example, in the German Bundesliga an order effect is distinctly necessary as it is well known that a team playing at its home ground has an advantage over the away team. If object-specific order effects are considered, one allows for different home effects across the teams.

To sum up all options how to specify components in the general paired comparison model (4), Table 2 collects all types of covariates and all possible parameterizations we consider.

## **3.3. Representation as generalized linear model**

It is straightforward to embed the considered models into the framework of generalized linear models (GLMs). The basic model (4) is a so-called cumulative ordinal regression model, which is a multivariate GLM. It has the form  $\log(\mathbb{P}(Y_{i(r,s)} \leq k)/\mathbb{P}(Y_{i(r,s)} > k)) = \eta_{i(r,s)k}$  with linear predictors

$$\eta_{i(r,s)k} = \delta + \theta_k + \gamma_{ir} - \gamma_{is}, \quad k = 1, \dots, K - 1.$$

Covariate type	Effect type	$\gamma_{ir} =$	$\gamma_{is} =$	$\eta_{i(rs)} =$
intercept	object-spec.	$\beta_{r0}$	$\beta_{s0}$	$\beta_{r0} - \beta_{s0}$
subject-spec. $\mathbf{x}_i$	object-spec.	$+\mathbf{x}_i^\top \boldsymbol{\beta}_r$	$+\mathbf{x}_i^\top \boldsymbol{\beta}_s$	$+\mathbf{x}_i^\top (\boldsymbol{\beta}_r - \boldsymbol{\beta}_s)$
subject-object-spec. $\mathbf{z}_{ir}$	global	$+\mathbf{z}_{ir}^\top \boldsymbol{\tau}$	$+\mathbf{z}_{is}^\top \boldsymbol{\tau}$	$+(\mathbf{z}_{ir} - \mathbf{z}_{is})^\top \boldsymbol{\tau}$
↳ incl. object-spec. $\mathbf{z}_r$	global	$+\mathbf{z}_r^\top \boldsymbol{\tau}$	$+\mathbf{z}_s^\top \boldsymbol{\tau}$	$+(\mathbf{z}_r - \mathbf{z}_s)^\top \boldsymbol{\tau}$
subject-object-spec. $\mathbf{z}_{ir}$	object-spec.	$+\mathbf{z}_{ir}^\top \boldsymbol{\alpha}_r$	$+\mathbf{z}_{is}^\top \boldsymbol{\alpha}_s$	$+\mathbf{z}_{ir}^\top \boldsymbol{\alpha}_r - \mathbf{z}_{is}^\top \boldsymbol{\alpha}_s$
order effect	global	$+\delta$		$+\delta$
order effect	object-spec.	$+\delta_r$		$+\delta_r$

Table 2: Overview over parameterizations of covariate effects available in **BTLasso**.

All the models considered in the previous section are also multivariate GLMs with more structured predictors. For example, in the case of subject-object-specific covariates the predictor has the form

$$\eta_{i(r,s)k} = \delta + \theta_k + \beta_{r0} - \beta_{s0} + (\mathbf{z}_{ir} - \mathbf{z}_{is})^\top \boldsymbol{\tau}, \quad k = 1, \dots, K - 1$$

or

$$\eta_{i(r,s)k} = \delta + \theta_k + \beta_{r0} - \beta_{s0} + \mathbf{z}_{ir}^\top \boldsymbol{\alpha}_r - \mathbf{z}_{is}^\top \boldsymbol{\alpha}_s, \quad k = 1, \dots, K - 1.$$

As far as estimation by maximum likelihood (ML) is concerned the only difference is in the structure of the linear predictor and the design matrix, which do not only consist of dummies for the objects as in the basic model.

Thus, in principle all the tools provided by GLMs, including tests, computation of standard errors, and algorithms for the maximization of the log-likelihood, are available. For the theory and computational tools, see, for example, [Fahrmeir and Tutz \(1997\)](#) or [Tutz \(2012\)](#).

## 4. Regularized estimation

In BTL models the inclusion of covariates yields models that can contain a huge number of parameters. If one includes, for example, a  $p$ -dimensional vector of subject-specific covariates one has to estimate a  $(p + 1)$ -dimensional parameter vector  $(\beta_{r0}, \boldsymbol{\beta}_r)$  for each object, which already for a moderate number of objects and a moderate size of the covariate vector yields a large number of parameters to estimate. Therefore, in applications ML estimates tend to be unstable or may not even exist.

One of the main features of package **BTLasso** is the selection of relevant terms and the implicit reduction of the complexity of the model. In the simplest case selection refers to the selection of variables, since typically in high-dimensional settings not all of the explanatory variables have an impact on the preference of objects. However, model complexity can also be reduced by identifying clusters of objects that share the same strength or variable effects. For the selection of relevant terms, we propose to use a penalized likelihood approach. Instead of maximizing the general log-likelihood  $l(\boldsymbol{\xi})$ , one maximizes the penalized log-likelihood

$$l_p(\boldsymbol{\xi}) = l(\boldsymbol{\xi}) - \lambda J(\boldsymbol{\xi}),$$

where  $l(\boldsymbol{\xi})$  is the usual log-likelihood with  $\boldsymbol{\xi}$  denoting a vector that contains all the parameters of the model.  $J(\boldsymbol{\xi})$  is a penalty term that penalizes specific structures in the parameter vector.

The parameter  $\lambda$  is a tuning parameter that specifies how seriously the penalty term has to be taken. For the extreme choice  $\lambda = 0$  one obtains the ML estimate. The choice of the tuning parameter is discussed later.

#### 4.1. Penalties

Whatever the type of the variables and the components included in the model, the threshold parameters  $\theta_k$  of the ordinal model are never penalized. For the other model components we propose fusion and selection penalties, which are given in the following.

##### *Clustering of intercept parameters*

The penalty

$$P_1(\beta_{10}, \dots, \beta_{m0}) = \sum_{r < s} |\beta_{r0} - \beta_{s0}|$$

aims at the identification of clusters of objects that share the same intercept parameter. For  $\lambda \rightarrow \infty$  one obtains one big cluster with  $\beta_{10} = \dots = \beta_{m0}$ . Due to the symmetric side constraint, that means that  $\beta_{10} = \dots = \beta_{m0} = 0$  and all intercept parameters are eliminated from the model. For an appropriately chosen tuning parameter  $\lambda$  ideally the clusters of objects that share equal intercepts are found. If no covariates are included the intercepts are equivalent to the total strength of the objects. Therefore, the penalty is a tool to identify the objects that really have to be distinguished. If explanatory variables are included the interpretation of the clusters refers to the strength or attractiveness left after accounting for the effects of the variables.

For pure object-specific covariates the attractiveness has the form  $\gamma_{ir} = \beta_{r0} + \mathbf{z}_{ir}^\top \boldsymbol{\tau}$ . In this case, the penalty provides a way to obtain unique parameter estimates despite the non-identifiability of the model. The penalty serves to find out how much of the attractiveness of an object is due to observed characteristics of the objects. When used in addition to the regular side constraint  $\sum_{r=1}^m \beta_{rj} = 0$  it restricts the intercepts and, therefore, the individual attractiveness that is left if one accounts for the object-specific covariates. If the tuning parameter gets large,  $\lambda \rightarrow \infty$ , all strength parameters  $\beta_{r0}$  are estimated as identical and the total strength is determined solely by  $\mathbf{z}_{ir}^\top \boldsymbol{\tau}$ . The corresponding model assumes that the abilities do depend on explanatory variables only. This restricted model was proposed by [Springall \(1973\)](#), however, it is hardly appropriate when a limited number of explanatory variables is available. Package **BTLLasso** can also be used to estimate the ranking lasso as proposed by [Masarotto and Varin \(2012\)](#). This is obtained by applying the penalty  $P_1$  described above to a setting without additional covariates.

##### *Clustering and selection of variable effects*

For object-specific effects, which are used for subject-specific covariates, the penalty

$$P_2(\beta_1, \dots, \beta_m) = \sum_{j=1}^{p_x} \sum_{r < s} |\beta_{rj} - \beta_{sj}|$$

yields clusters of objects that share the same effect for the single variables (denoted by  $j$ ). Let us consider the variable gender in the preference for parties. If  $\beta_{rj}$  is the same for  $r \in R$ , comparisons between parties from a set of parties  $R$  do not depend on gender. However,

if  $\beta_{rj} = \beta_{\cdot j}^{(1)}$  for  $r \in R^{(1)}$  and  $\beta_{rj} = \beta_{\cdot j}^{(2)}$  for  $r \in R^{(2)}$ , gender distinguishes between the groups of parties  $R^{(1)}$  and  $R^{(2)}$ . As in the case of the intercept parameters, side constraints like  $\sum_{r=1}^m \beta_{rj} = 0$  for every covariate  $j$  are needed. If  $\lambda$  increases successively all effects corresponding to single covariates are merged to one cluster and the corresponding covariates are eliminated from the model as all effects become zero. For  $\lambda$  small enough some of the covariates are deleted and for the others clusters of objects that share the same effect are identified.

For global effects, which are used for object-specific or subject-object-specific covariates, one uses the penalty

$$P_3(\boldsymbol{\tau}) = \sum_{j=1}^{p_2} |\tau_j|.$$

It is a simple lasso penalty that restricts the absolute values of the parameters. For  $\lambda \rightarrow \infty$  all corresponding covariates are eliminated from the model.

For object-specific parameters that are used on subject-object-specific covariates, we use the modified penalty

$$P_4(\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_m) = \sum_{j=1}^{p_1} \sum_{r < s} |\alpha_{rj} - \alpha_{sj}| + \nu_1 \sum_{j=1}^{p_1} \sum_{r=1}^m |\alpha_{rj}|, \quad \nu_1 \in \{0, 1\}.$$

For this kind of parameterization, no further side constraints are needed. Therefore, the respective penalty is composed of two parts. The first part penalizes all absolute pairwise differences between the coefficients corresponding to one covariate, the second part (optionally for  $\nu_1 = 1$ ) penalizes the absolute values of all parameters. By specifying  $\nu_1$ , the user has the option to decide if only clustering of the objects with regard to the covariates is intended ( $\nu_1 = 0$ ) or if, additionally, selection of the parameters should be enabled ( $\nu_1 = 1$ ). For  $\lambda \rightarrow \infty$ , the specification of  $\nu_1$  leads to different results. While for  $\nu_1 = 0$  one ends up with global effects (equal for all objects but not zero), for  $\nu_1 = 1$  all respective covariates are eliminated from the model.

#### *Clustering and selection of order effects*

The order effects are penalized by the terms

$$P_5(\delta) = |\delta|,$$

$$P_6(\delta_1, \dots, \delta_m) = \sum_{r < s} |\delta_r - \delta_s| + \nu_2 \sum_{r=1}^m |\delta_r|, \quad \nu_2 \in \{0, 1\},$$

depending on whether a global order effect  $\delta$  is specified or an object-specific order effect  $\delta_r$ . The first penalty  $P_5$  is used if one starts with one order effect that does not vary over objects, and only serves the purpose to restrict the size of the effect, which, in particular, may be zero. Penalty  $P_6$  for object-specific order effects allows to identify clusters of objects that share the same order effect. Additionally, the parameter  $\nu_2$  allows for a decision between solely clustering the order effects ( $\nu_2 = 0$ ) or additional selection of the order effects ( $\nu_2 = 1$ ). With growing tuning parameter  $\lambda$ , for  $\nu_2 = 0$  one ends up with one global order effect while for  $\nu_2 = 1$  all order effects can be eliminated from the model.

Covariate type	Effect type	Penalty
intercept	object-spec.	$P_1(\beta_{10}, \dots, \beta_{m0}) = \sum_{r < s}  \beta_{r0} - \beta_{s0} $
subject-spec. $\mathbf{x}_i$	object-spec.	$P_2(\beta_1, \dots, \beta_m) = \sum_{j=1}^{p_x} \sum_{r < s}  \beta_{rj} - \beta_{sj} $
subject-object-spec. $\mathbf{z}_{ir}$	global	$P_3(\boldsymbol{\tau}) = \sum_{j=1}^{p_2}  \tau_j $
↳ incl. object-spec. $\mathbf{z}_r$	global	$P_3(\boldsymbol{\tau}) = \sum_{j=1}^{p_2}  \tau_j $
subject-object-spec. $\mathbf{z}_{ir}$	object-spec.	$P_4(\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_m) = \sum_{j=1}^{p_1} \sum_{r < s}  \alpha_{rj} - \alpha_{sj}  + \nu_1 \sum_{j=1}^{p_1} \sum_{r=1}^m  \alpha_{rj} $
order effect	global	$P_5(\delta) =  \delta $
order effect	object-spec.	$P_6(\delta_1, \dots, \delta_m) = \sum_{r < s}  \delta_r - \delta_s  + \nu_2 \sum_{r=1}^m  \delta_r $

Table 3: Overview over all penalty terms on covariate effects available in **BTLasso**.

## 4.2. Combining penalties

Table 3 shows an overview on all presented penalty terms together with the corresponding type of covariate. All penalty terms can use internal weights according to the principle of adaptive lasso (Zou 2006) using the option `adaptive = TRUE` from the function `ctrl.BTLasso()`. For the sake of simplicity the adaptive weights have been omitted in the presentation of the penalty terms. Finally, all different penalty terms are combined using

$$J(\boldsymbol{\xi}) = \sum_{l=1}^L \psi_l P_l,$$

where  $\psi_l$  are penalty-specific weights. Although different tuning parameters for the different penalty terms are conceivable, the optimization of tuning parameters would become overly complicated. Therefore, all penalty terms are combined into one joint penalty controlled by the joint tuning parameter  $\lambda$ . In order to allow for comparability of the different penalty terms, two requirements have to be fulfilled. First, all covariates have to be transformed into comparable scales. For that purpose,

- the subject-specific covariates are scaled so that each vector  $\mathbf{x}_{.1}, \dots, \mathbf{x}_{.p_x}$  referring to one covariate across all subjects has variance 1;
- the subject-object-specific covariates are scaled so that each vector  $\mathbf{z}_{..1}, \dots, \mathbf{z}_{..p_1}$  referring to one covariate across all subjects and all objects has variance 1;
- the object-specific covariates are scaled so that each covariate vector  $\mathbf{z}_{.1}, \dots, \mathbf{z}_{.p_2}$  referring to one covariate across all objects has variance 1.

By default, the scaling listed above is executed automatically and can be controlled by the argument `scale` in `ctrl.BTLasso()`. It should be noted that all `print()` and `plot()` commands include the option `rescale` to specify whether the estimates should be rescaled to the original scale or not. While the rescaled estimates can be interpreted directly with

respect to the corresponding covariates, only the scaled parameters can be compared to each other in terms of effect sizes.

Second, the different penalty terms have to be weighted with weights  $\psi_l$  according to the number of penalties and according to the number of free parameters they include, following the principle of weighting penalties proposed by [Bondell and Reich \(2009\)](#) and [Oelker and Tutz \(2017\)](#). The weights are applied if `weight.penalties = TRUE` in `ctrl.BTLLasso()` and are multiplied with the (adaptive lasso) weights if `adaptive = TRUE`. The optimal tuning parameter is chosen by  $k$ -fold cross-validation which is described in more detail in the following section.

### 4.3. Implementation

In package **BTLLasso**, penalized Fisher scoring is used to fit the proposed models. Generally, all implemented penalties listed in [Table 3](#) are  $L_1$  penalties and, therefore, are not differentiable. However, following the suggestions of [Fan and Li \(2001\)](#) and [Oelker and Tutz \(2017\)](#)  $L_1$  penalties can be approximated by quadratic terms. Quadratic penalty terms are differentiable and can, therefore, easily be incorporated in a (penalized) Fisher scoring algorithm. A first implementation of quadratically approximated  $L_1$  penalties can be found in the R package **gvcn.cat** ([Oelker 2015](#)), but the package does not contain an implementation for cumulative logit models. For shorter computation time, the fitting algorithm itself is implemented in C++ and integrated into R using the packages **Rcpp** ([Eddelbuettel and François 2011](#); [Eddelbuettel 2013](#)) and **RcppArmadillo** ([Eddelbuettel and Sanderson 2014](#)).

## 5. Use of package BTLLasso

The two main functions of the package **BTLLasso** are the functions

```
BTLLasso(Y, X = NULL, Z1 = NULL, Z2 = NULL, lambda = NULL,
         control = ctrl.BTLLasso(), trace = TRUE)
```

and

```
cv.BTLLasso(Y, X = NULL, Z1 = NULL, Z2 = NULL, folds = 10,
            lambda = NULL, control = ctrl.BTLLasso(), cores = folds, trace = TRUE,
            trace.cv = TRUE, cv.crit = c("RPS", "Deviance"))
```

which carry out the penalized estimation of the specified model for a pre-specified grid of tuning parameters  $\lambda$ . The grid of tuning parameters can be specified via the argument `lambda` or, if `lambda = NULL`, is created automatically so that the maximal value of  $\lambda$  sets all penalized values or differences to exactly zero. Compared to `BTLLasso()`, `cv.BTLLasso()` additionally performs cross-validation to find the optimal value for the tuning parameter. The function `ctrl.BTLLasso()` contains many different options to control the model fit.

Of course, also simple Bradley-Terry models without covariates as described in [Section 2](#) can be computed with package **BTLLasso**. Therefore, package **BTLLasso** can also be a useful tool for the computation of Bradley-Terry models with ordinal response or for models with (possibly object-specific) order effects. For that purpose, all arguments to supply covariates (`X`, `Z1` and `Z2`) and all penalties can be ignored.

Argument	Covariate type	Effect type	Parameter	Penalty
<b>X</b>	subject-spec. $\mathbf{x}_i$	object-spec.	$\beta_{rj}$	$P_2$
<b>Z2</b>	subject-object-spec. $\mathbf{z}_{ir}$	global	$\tau_j$	$P_3$
	↳ incl. object-spec. $\mathbf{z}_r$	global	$\tau_j$	$P_3$
<b>Z1</b>	subject-object-spec. $\mathbf{z}_{ir}$	object-spec.	$\alpha_{rj}$	$P_4$

Table 4: Overview over the three possible arguments to supply covariates in **BTLasso**.

### 5.1. Model specification

In the following, it is described which functions and arguments are needed to specify a paired comparison model with package **BTLasso**. More details and illustrations can be found in Section 6 where specific examples based on exemplary data sets are presented.

#### *Response*

Both in `BTLasso()` and `cv.BTLasso()`, the response (i.e., the paired comparisons) is specified using the argument `Y`. To create a response object the function `response.BTLasso()` is used where one needs to declare the arguments `response`, `first.object`, `second.object` and (possibly) `subject`. Only if a subject is specified, the paired comparisons can be related to subject-specific or subject-object-specific covariates. To facilitate the compatibility of the package with the `psychotree` package, `response.BTLasso()` can also take a ‘`paircomp`’ object (the standard object class used in `psychotree`) as its only argument (argument `response`). Then, all necessary information is extracted automatically and stored in a ‘`response.BTLasso`’ object. The argument `order.vec` is explained in the paragraph on order effects.

#### *Covariates*

In the functions `BTLasso()` and `cv.BTLasso()`, three different arguments are provided to specify all covariates that can be incorporated in a model, namely the arguments `X`, `Z1` and `Z2`. The argument `X` contains all subject-specific covariates  $\mathbf{x}_i$ , the number of subject-specific covariates (the number of columns of `X`) is denoted by  $p_x$ . In contrast, `Z1` and `Z2` contain object-specific covariates. In particular, `Z2` contains all object-specific covariates  $\mathbf{z}_r$  or subject-object-specific covariates  $\mathbf{z}_{ir}$  which are modeled with global parameters  $\boldsymbol{\tau}$ . In total, `Z2` contains  $p_2$  covariates. `Z1` contains  $p_1$  subject-object-specific covariates  $\mathbf{z}_{ir}$  modeled with object-specific parameters  $\boldsymbol{\alpha}_r$ . Table 4 contains an overview over all possible assignments of covariates.

#### *Order effects*

In principle, the order effect (home advantage in sport events) could also be considered to be a special case of a subject-object-specific covariate. In the example of Bundesliga matches, the teams are considered to be the objects and the match days are considered to be the subjects. Therefore, one can create a subject-object-specific covariate  $z_{ir}$  representing the home effect, either using effect coding or dummy coding. With effect coding, one specifies  $z_{ir} = 1$  if team  $a_r$  is the home team on match day  $i$  and  $z_{is} = -1$  for the away team  $a_s$ , for all other teams one sets  $z_{it} = 0$ ,  $t \notin \{r, s\}$ . With dummy coding, one specifies  $z_{ir} = 1$  if team  $a_r$  is the home

team on match day  $i$  and  $z_{it} = 0$ ,  $t \neq r$ , otherwise. As subject-object-specific covariates, order effects can either be parameterized with a global parameter  $\delta$  (global home effect) or with a team-specific effect  $\delta_r$  (team-specific home effect). Technically order effects could be integrated into the modeling framework as subject-object-specific covariates. Nevertheless, they play a special role in the modeling of paired comparisons. Even if no characteristics of objects or subjects are available one might want to include order effects when modeling data because ignoring the presence of an order effect may lead to a severe bias in the estimates. Then one implicitly uses subject-object-specific covariates, maybe without being aware of using this type of covariate.

In package **BTLLasso**, order effects are already implemented and can be applied using the arguments `order.effect` and `object.order.effect` from the `ctrl.BTLLasso()` function for global and object-specific order effects, respectively. In the case of object-specific order effects, the option `order.center = TRUE` allows for the use of effect coding instead of dummy coding in the design matrix. In specific cases it might occur that an order effect is present only in some of the paired comparisons while being absent in the other paired comparisons. For example, in the UEFA Champions League all matches are played in the home stadium of one of the teams with the exception of the final match (which is played on neutral ground). To account for such a case, the function `response.BTLLasso` contains the argument `with.order`. This argument allows to specify a Boolean vector (of the same length as the response) that contains the information which paired comparisons have an order effect.

### *Intercepts*

By default, object-specific intercepts are included in the model. To eliminate the intercepts, the option `include.intercepts = FALSE` from `ctrl.BTLLasso()` can be used.

## 5.2. Specification of penalties

In Section 4.1 several possible penalties were proposed which can be applied to reduce the complexity of the model and to make the model easier to interpret. All these penalties can be applied to the specified model using specific options within `ctrl.BTLLasso()`. In particular, the user can specify the following arguments:

- `penalize.intercepts`: (de-)activates penalty  $P_1$ .
- `penalize.X`: (de-)activates penalty  $P_2$ .
- `penalize.Z2`: (de-)activates penalty  $P_3$ .
- `penalize.Z1.diffs`: (de-)activates penalty  $P_4$ .
- `penalize.Z1.absolute`: sets  $\nu_1 = 1$  in  $P_4$ .
- `penalize.order.effect.diffs`: (de-)activates penalty  $P_6$ .
- `penalize.order.effect.absolute`: (de-)activates penalty  $P_5$  or sets  $\nu_2 = 1$  in  $P_6$ .

The arguments can either be set to `TRUE` or `FALSE` and, thereby, (de-)activate the respective penalties. The arguments `penalize.X`, `penalize.Z1.diffs`, `penalize.Z.absolute` and `penalize.Z2` can also be used differently. Instead of setting them to `TRUE` or `FALSE` they can

also take character vectors as input. Then, the character vectors contain the variable names corresponding to the parameters that should be penalized. As an example, we look at the GLES data where  $X$  consists of the three variables `Age`, `Gender` and `Abitur`. Then, specifying `penalize.X = c("Age", "Gender")` prevents the parameters corresponding to `Abitur` from being penalized. The arguments `penalize.X = c("Age", "Gender", "Abitur")` and `penalize.X = TRUE` are equivalent.

### 5.3. Cross-validation

To perform cross-validation of the specified model the function `cv.BTLLasso()` is used. By default, this is done as a 10-fold cross-validation (`fold = 10`) and is parallelized on several cores using the argument `cores`. One can choose between two different options for the cross-validation criterion, namely the ranked probability score (RPS) and the deviance. For ordinal response  $y \in \{1, \dots, K\}$  the RPS (Gneiting and Raftery 2007) is recommended which can be denoted by

$$\text{RPS}(y, \hat{\pi}(k)) = \sum_{k=1}^K (\hat{\pi}(k) - \mathbf{1}(y \leq k))^2,$$

where  $\pi(k)$  represents the cumulative probability  $\pi(k) = \mathbb{P}(y \leq k)$ . In contrast to the deviance, the RPS takes advantage of the ordinal structure of the response.

The advantage of cross-validation over information based selection criteria is that it avoids the problem to determine the effective degrees of freedom. However, in specific data settings, in particular if no covariates are involved, it might be necessary to adapt the number of folds. If one fits, for example, a classical Bradley-Terry model, in which only strength parameters are estimated, employing cross-validation may be problematic since one might remove all competitions in which a specific object is involved. Then the strength parameter of this object cannot be estimated. Let us consider in more detail the case of a Bradley-Terry model with order effects in a complete design. Then the number of pairwise comparisons is  $m(m-1)$  and each object occurs in  $2(m-1)$  paired comparisons. If one uses 10-fold cross-validation  $m(m-1)/10$  paired comparisons are left out. Then the critical case, in which the strength parameter of an object cannot be estimated, can occur if  $m(m-1)/10 > 2(m-1)$ , or equivalently,  $m/10 > 2$ . For small to moderate  $m$  this will not happen. For example, if  $m = 10$  one has 90 paired comparisons and each object is involved in 18 comparisons. When using 10-fold cross-validation one leaves out 9 paired comparisons, which means that for each object at least 9 paired comparisons are left. However, if  $m$  is large and therefore  $m/10 > 2$  holds not all parameters might be estimable. If, for example,  $m = 30$  there is some danger that some parameters cannot be estimated. Then one has to increase the number of folds. The problem occurs in particular for large  $m$ , which is the rare case in paired comparison models. Nevertheless, we return an error message whenever one of the parameters is not estimable. Then, the user is asked to use another starting value or to increase the number of folds for cross-validation.

### 5.4. Bootstrap intervals

To receive bootstrap intervals for the parameters a bootstrap method is implemented in the function `boot.BTLLasso()`. In each of the  $B$  bootstrap iterations the method is applied to a data set randomly sampled from the original data set. In every iteration the optimal tuning

parameter is determined separately by cross-validation to fully account for the additional variability induced by the cross-validation. Therefore, the bootstrap procedure can be very time-consuming, although parallelization is possible using the argument `cores`. In general, the model generated by `cv.BTLLasso()` is supplied to `boot.BTLLasso()`. To speed up computation of the bootstrap, additionally a new (shorter) vector of tuning parameters `lambda` can be supplied. Finally, from the bootstrap estimates the requested empirical quantiles are calculated and bootstrap intervals can be obtained and visualized by a `print()` and a `plot()` method.

## 5.5. Visualization and supportive methods

In addition to the aforementioned `plot()` method for empirical bootstrap intervals, two further functions for visualization of the results are provided. The main visualization tool is the `plot()` method for ‘`BTLLasso`’ and ‘`cv.BTLLasso`’ objects. This method plots the coefficient paths of every single estimate produced both by `BTLLasso()` and `cv.BTLLasso()` along the tuning parameter. With `plot()`, for every covariate (and for intercepts and order effects), separate plots are generated. These plots are especially helpful to investigate the clustering behavior of the penalties along the tuning parameter and to investigate different covariate effects across objects. In contrast, the second visualization function called `paths()` generates one path per covariate and, therefore, only generates a single plot containing all paths. The paths either visualize the penalty terms for the single model components (`y.axis = "penalty"`) or the  $L_2$  norm (`y.axis = "L2"`) of the vector of all respective parameters, e.g., the  $L_2$  norm of all parameters corresponding to one covariate. This type of visualization helps to compare effect sizes or variable importance between different variables or model components. If a cross-validated model is supplied instead of the regular model, additionally the optimal model according to cross-validation is highlighted. Both visualization techniques will be illustrated in the following section.

In addition to these visualization methods, the package also provides several further supportive methods to analyze and process the results of the main functions of the package. First of all, for all objects generated by `boot.BTLLasso()`, `BTLLasso()` or `cv.BTLLasso()` also `print()` methods exist to print the most important results of the respective objects into the console. Beside that one can apply `coef()` and `logLik()` to ‘`cv.BTLLasso`’ objects. A very helpful method is `predict()` which can be applied both to ‘`BTLLasso`’ and ‘`cv.BTLLasso`’ objects. The argument `type` can be specified to be `type = "link"` to return the linear predictors, `type = "response"` to return probabilities for the single response categories and `type = "trait"` to return the traits/abilities separately for both competing objects. `predict()` can be applied both to new data and to the original data used to estimate the model.

## 6. Applications

In the following section, the presented methods are applied to the exemplary data sets of the German election study and the German Bundesliga as described in Section 3.1.

### 6.1. Election data

In the election data, both subject-specific and subject-object-specific covariates are available.

In contrast to [Schauberger and Tutz \(2017\)](#), where only subject-specific covariates are considered, here both types of covariates are included in the model simultaneously. We choose to allow for subject-specific effects for the subject-object-specific covariates `SocEc`, `Immigration` and `Climate` which have to be supplied to `cv.BTLLasso()` via the `Z1` argument. The subject-specific covariates `Age`, `Gender` and `Abitur` are supplied via `X`. The following code is used to fit the optimal cross-validated model to our data:

```
R> data("GLES", package = "BTLLasso")
R> Y <- GLES$Y
R> X <- scale(GLES$X[, c(1, 2, 5)], scale = FALSE)
R> Z1 <- scale(GLES$Z1, scale = FALSE)
R> lambda <- exp(seq(log(201), log(1), length = 100)) - 1
R> set.seed(1860)
R> mcv_GLES <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1, lambda = lambda)
```

For our model, we wish to penalize both the covariates included in `X` and `Z1`. Furthermore, no order effect is necessary as the order of the parties is random in this application. These model assumptions match the default settings in `ctrl.BTLLasso()` and, therefore, no changes have to be applied to the `control` argument.

Subsequently, the estimated parameter paths can be plotted using the commands

```
R> plot(mcv_GLES, which = 5:7, plots_per_page = 3)
R> plot(mcv_GLES, which = 2:4, plots_per_page = 3)
```

separately for all covariates as depicted in [Figure 1](#). Here, `plots_per_page` specifies the number of plots placed on one page. The default is `plots_per_page = 1` where each model component is placed on a separate page, in an interactive R session each page is switched by hand. The `which` argument specifies the model component to plot, we omitted here the first component which are the (unpenalized) intercepts. In [Figure 1](#), the optimal model according to the 10-fold cross-validation is highlighted by the red dashed line.

The parameter paths are drawn along the tuning parameter or rather  $\log(\lambda + 1)$ . On first sight, a clear difference between the subject-object-specific covariates in [Figure 1a](#) and the subject-specific covariates in [Figure 1b](#) becomes obvious. Beside the restriction implied by the penalty, the coefficients in [Figure 1a](#) are not restricted while the coefficients in [Figure 1b](#) have to sum up to zero for every covariate. Therefore, the estimates for subject-object-specific covariates like the attitude towards climate change can be different from zero but still equal for all parties, which is not possible (and not sensible) for variables like `Gender` or `Age`.

It can be seen, that (according to the red dashed line indicating the optimal model according to cross-validation) a rather non-sparse model is chosen. Most of the possible covariate effects differ across parties. For the subject-object-specific variables, all effects are negative. This result is very intuitive, it simply states that for a higher (self-perceived) distance of voter and party on a specific issue the attractiveness of the party for the voter decreases. For the socio-economic dimension, three clusters of parties are distinguished, namely {FDP, SPD, Greens}, {CDU/CSU} and {Left Party}. In that case a cluster means, that for these parties the distance between party and voter is equally important, it is most important for the attractiveness of the Left Party and less important for the attractiveness of FDP, SPD and Greens. Comparing the effect sizes, the issue of immigration has the lowest effect. For the climate issue, the

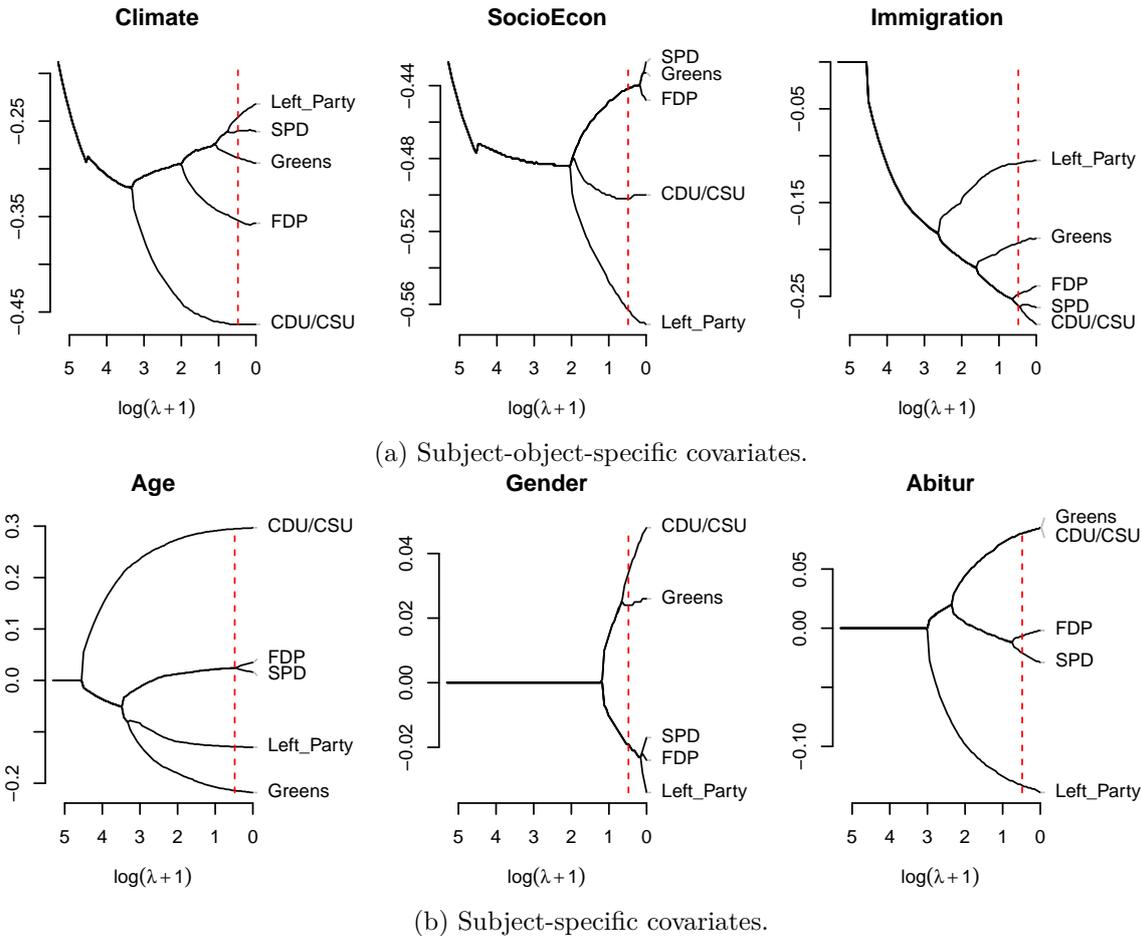


Figure 1: Parameter paths for subject-object-specific (a) and subject-specific (b) variables of the GLES data. The dashed red lines represent the optimal model according to 10-fold cross-validation.

differences between the parties are rather high. Also for the subject-specific covariates rather many different clusters are found for the single covariates, four clusters for **Age** and **Abitur** and three clusters for **Gender**. For example, the attractiveness of CDU/CSU is increased with growing age while the attractiveness of Greens and Left Party decreases.

Figure 2 was created by the command

```
R> paths(mcv_GLES, y.axis = "L2")
```

and shows the  $L_2$  norms of all covariates along the tuning parameter. The  $L_2$  norm measures the size of each vector which collects all estimates corresponding to one covariate. Therefore, it can be seen as measure of variable importance as long as the estimates for the scaled covariates are used. It can be seen that the subject-object-specific covariates representing the different political dimensions are more important than the subject-specific variables **Age**, **Gender** and **Abitur**.

To conclude the analysis of the election data, we present how to generate empirical bootstrap intervals. The bootstrap estimation is executed using the following command:

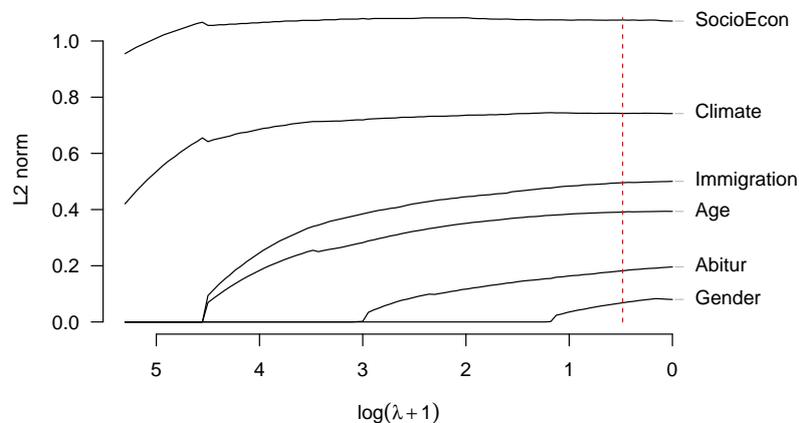


Figure 2:  $L_2$  paths for both subject-object-specific and subject-specific variables of the GLES data. The dashed red line represents the optimal model according to 10-fold cross-validation.

```
R> set.seed(1860)
R> mboot_GLES <- boot.BTLLasso(mcv_GLES, B = 500, cores = 25)
```

The main argument of the function is the object generated from cross-validation. Furthermore, `B` specifies the number of bootstrap samples and `cores` specifies the number of cores to use. The commands

```
R> plot(mboot_GLES, which = 5:7, plots_per_page = 3)
R> plot(mboot_GLES, which = 2:4, plots_per_page = 3)
```

are used to visualize the empirical bootstrap intervals which are shown in Figure 3.

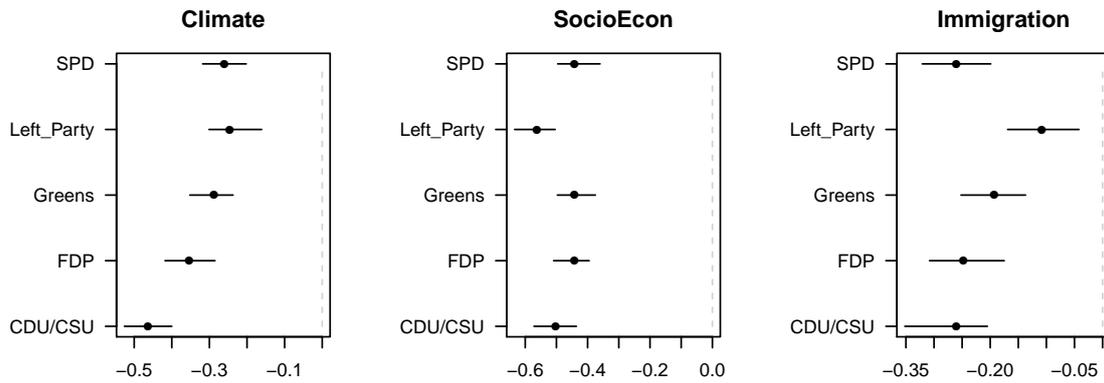
In general, the bootstrap intervals illustrate the variability of the single estimates and provide hints if single estimates are distinctly different or not. For example, the bootstrap intervals for `Age` show that for this variable distinct differences exist between the single parties while this seems not to be the case for `Gender`.

## 6.2. Football data

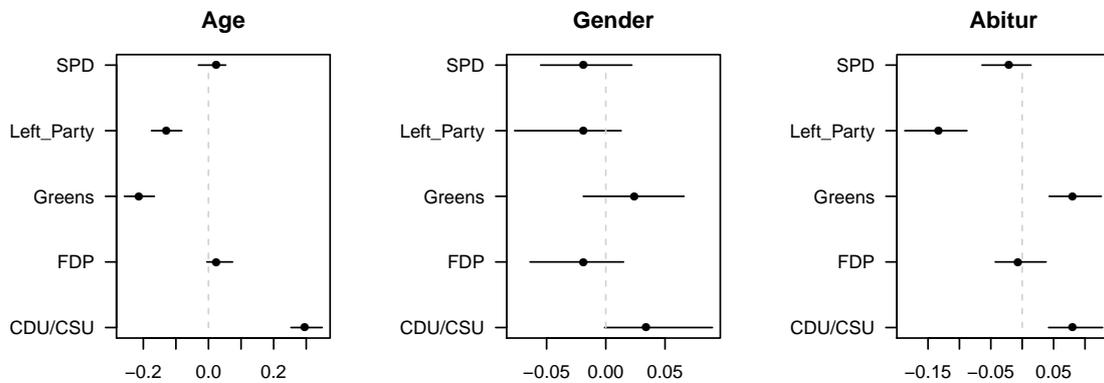
For the data from the German Bundesliga, two different models are fitted for illustration. In the first model, the subject-object-specific variables `Distance` and `BallPossession` are used to investigate if these variables influence the strength of the different football teams. In the second model, the influence of the object-specific variable market value is investigated.

The first model in total contains four different model components, namely team-specific home effects (i.e., subject-specific order effects), intercepts and team-specific effects for both subject-object-specific variables `Distance` and `BallPossession`. The model is computed using the following code:

```
R> data("Buli1516", package = "BTLLasso")
R> Y <- Buli1516$Y5
R> Z1 <- scale(Buli1516$Z1[, 1:(2 * 18)], scale = FALSE)
```



(a) Subject-object-specific covariates.



(b) Subject-specific covariates.

Figure 3: Bootstrap intervals for subject-object-specific (a) and subject-specific (b) variables of the GLES data according to 500 bootstrap samples.

```
R> ctrl_buli <- ctrl.BTLLasso(object.order.effect = TRUE,
+   name.order = "Home", penalize.order.effect.diffs = TRUE,
+   penalize.order.effect.absolute = FALSE, order.center = TRUE)
R> set.seed(1860)
R> mcv_buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl_buli)
```

The decisive part is the definition of the control argument using `ctrl.BTLLasso()`. In football matches, unarguably a home effect is present which is equivalent to the order effect in paired comparisons because the first-named team is the home team. As we do not doubt the general presence of a home effect, the absolute values of the home effects of the single teams are not penalized (`penalize.order.effect.absolute = FALSE`). Only the differences between the single home effects are penalized (`penalize.order.effect.diffs = TRUE`) to find out, whether all teams share the same home effect or if (clusters of) single teams have distinct home effects. In addition, the covariates in `Z1` are centered for a better interpretability of the intercepts (which are not penalized).

Figure 4 shows the coefficient paths for the four model components along  $\log(\lambda + 1)$  created by `plot()`. It can be seen that the optimal model in this case is very sparse. All teams share

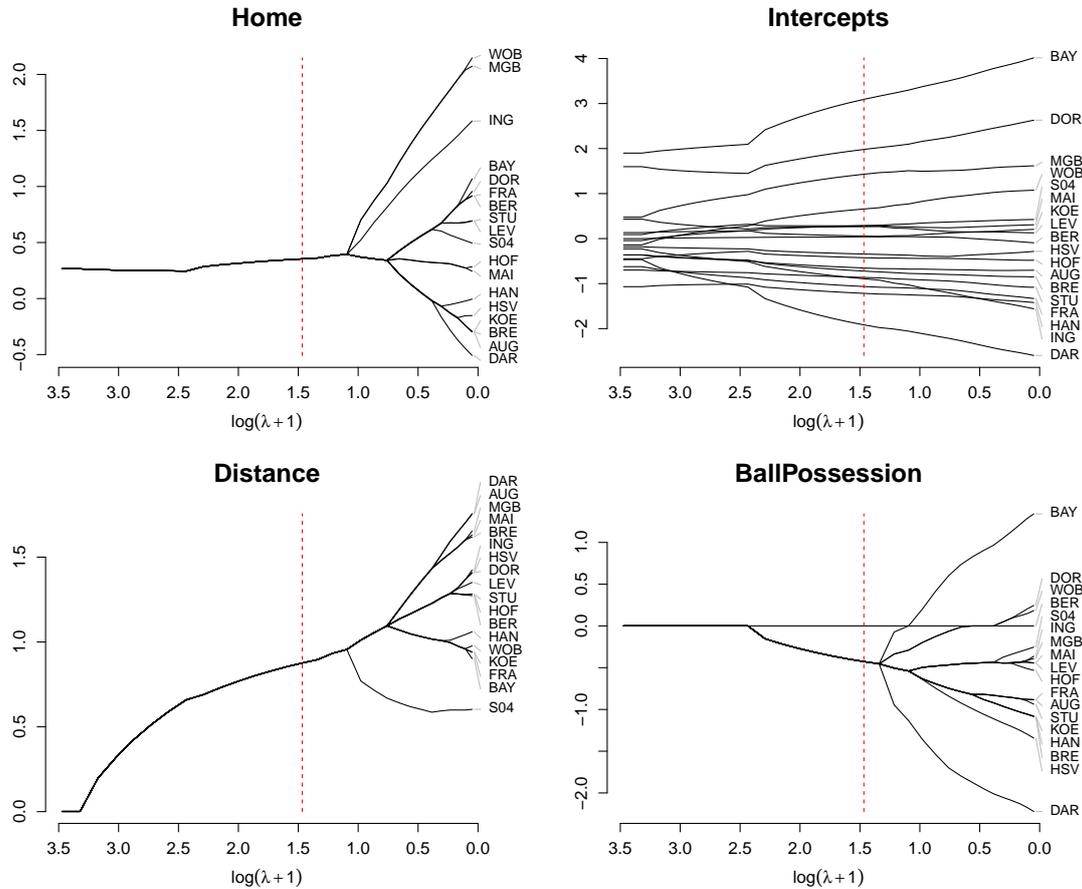


Figure 4: Coefficient paths for the model with subject-object-specific covariates `Distance` and `BallPossession` and (possibly) team-specific home effects. The red dashed line represents the optimal model according to 10-fold cross-validation.

the same home effect. Furthermore, all teams share the same, quite strong positive effect of `Distance`. That means, a strong running performance is strongly associated with a better performance of a team. For `BallPossession`, most teams (except for Hertha BSC Berlin) share the same negative effect. Therefore, higher values in `BallPossession` are associated with rather worse results. Overall, the path plots illustrate the behavior of the penalty terms. Both for the home effects and for the effects for `Distance` and `BallPossession` the penalty forms clusters of teams with equal effects. With decreasing values of the tuning parameter, the clusters are decomposed into smaller clusters ending up with singleton clusters for each team.

The second model contains only three model components, namely the home effects, the intercepts and the effect of the object-specific covariate market value. The model is computed using the following code:

```
R> data("Buli1516", package = "BTLLasso")
R> Y <- Buli1516$Y5
R> Z2 <- Buli1516$Z2
```

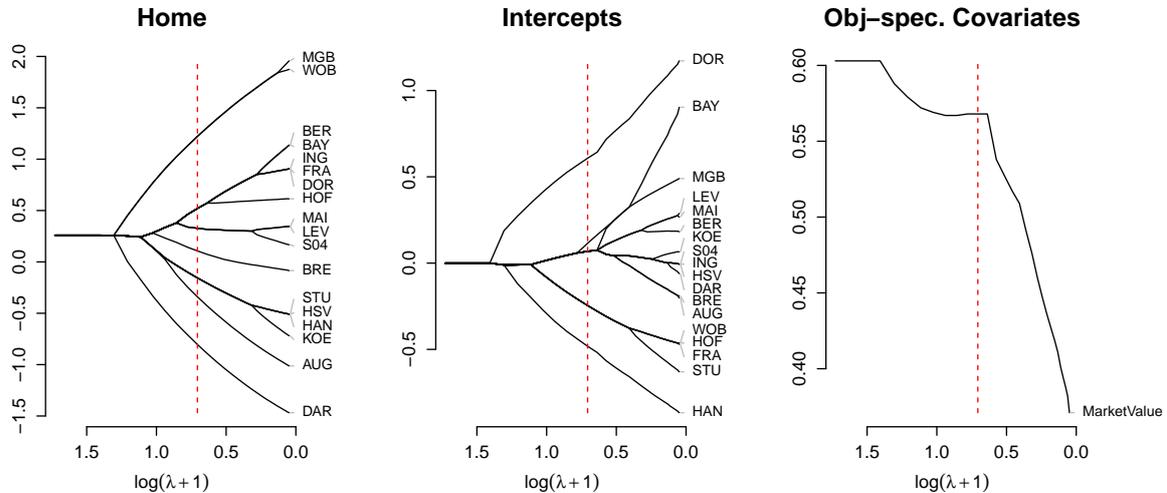


Figure 5: Coefficient paths for the model with object-specific covariate market value, penalized intercepts and (possibly) team-specific home effects. The red dashed line represents the optimal model according to 10-fold cross-validation.

```
R> ctrl.buli2 <- ctrl.BTLLasso(object.order.effect = TRUE,
+   name.order = "Home", penalize.order.effect.diffs = TRUE,
+   penalize.order.effect.absolute = FALSE, order.center = TRUE,
+   penalize.intercepts = TRUE)
R> set.seed(1860)
R> mcv_buli2 <- cv.BTLLasso(Y = Y, Z2 = Z2, control = ctrl.buli2)
```

In contrast to the previous model, now the intercepts are also penalized due to the identifiability issues with object-specific covariates described in Section 3.2. Therefore, the question answered by the model is whether the market value (which is not penalized) is sufficient to explain the match outcomes or if, additionally, home effects and intercepts are necessary for (clusters of) single teams.

Figure 5 shows the coefficient paths for this model created by `plot()`. Obviously, the market value has a positive effect for the performance of a team. Both for the home effects and for the intercepts, in the optimal model several clusters occur. For the home effects, seven different clusters evolve with {MGB, WOB} and {DAR} having the best and worst home effect, respectively. For the intercepts, five clusters are found with Borussia Dortmund forming a cluster of its own with the highest value.

## 7. Summary

The R package **BTLLasso** presented in this work provides a general framework for the inclusion of different kinds of covariates in paired comparison models. Covariates can vary over subjects or objects or both over subjects and objects of paired comparisons. In addition, (possibly object-specific) order effects can be included. The paired comparisons can be both binary and ordinal. To obtain sparser models and easier interpretation, suitable penalty terms for all model components are provided. For all model components with object-specific effects,

the absolute pairwise differences of the parameters can be penalized to force clustering of objects with respect to the corresponding model component. The package contains functions for cross-validation, bootstrap intervals and for the visualization of results. The package also provides exemplary data sets that illustrate the possible applications of **BTLasso**.

## References

- Agresti A (1992). “Analysis of Ordinal Paired Comparison Data.” *Journal of the Royal Statistical Society C*, **41**(2), 287–297. doi:10.2307/2347562.
- Bondell HD, Reich BJ (2009). “Simultaneous Factor Selection and Collapsing Levels in ANOVA.” *Biometrics*, **65**(1), 169–177. doi:10.1111/j.1541-0420.2008.01061.x.
- Bradley RA (1976). “Science, Statistics, and Paired Comparison.” *Biometrics*, **32**(2), 213–232. doi:10.2307/2529494.
- Bradley RA, Terry ME (1952). “Rank Analysis of Incomplete Block Designs, I: The Method of Pair Comparisons.” *Biometrika*, **39**(3/4), 324–345. doi:10.2307/2334029.
- Casalicchio G (2014). *ordBTL: Modelling Comparison Data with Ordinal Response*. R package version 0.8, URL <https://CRAN.R-project.org/package=ordBTL>.
- Casalicchio G, Tutz G, Schauburger G (2015). “Subject-Specific Bradley-Terry-Luce Models with Implicit Variable Selection.” *Statistical Modelling*, **15**(6), 526–547. doi:10.1177/1471082x15571817.
- Cattelan M (2012). “Models for Paired Comparison Data: A Review with Emphasis on Dependent Data.” *Statistical Science*, **27**(3), 412–433. doi:10.1214/12-sts396.
- David HA (1988). *The Method of Paired Comparisons*. 2nd edition. Griffin, London.
- Davidson RR (1970). “On Extending the Bradley-Terry Model to Accommodate Ties in Paired Comparison Experiments.” *Journal of the American Statistical Association*, **65**(329), 317–328. doi:10.1080/01621459.1970.10481082.
- Dittrich R, Francis B, Hatzinger R, Katzenbeisser W (2007). “A Paired Comparison Approach for the Analysis of Sets of Likert-Scale Responses.” *Statistical Modelling*, **7**(1), 3–28. doi:10.1177/1471082x0600700102.
- Dittrich R, Hatzinger R, Katzenbeisser W (2002). “Modelling Dependencies in Paired Comparison Data: A Log-Linear Approach.” *Computational Statistics & Data Analysis*, **40**(1), 39–57. doi:10.1016/s0167-9473(01)00106-2.
- Eddelbuettel D (2013). *Seamless R and C++ Integration with Rcpp*. Springer-Verlag, New York.
- Eddelbuettel D, François R (2011). “**Rcpp**: Seamless R and C++ Integration.” *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

- Eddelbuettel D, Sanderson C (2014). “**RcppArmadillo**: Accelerating R with High-Performance C++ Linear Algebra.” *Computational Statistics & Data Analysis*, **71**, 1054–1063. doi:[10.1016/j.csda.2013.02.005](https://doi.org/10.1016/j.csda.2013.02.005).
- Fahrmeir L, Tutz G (1997). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer-Verlag, New York.
- Fan J, Li R (2001). “Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties.” *Journal of the American Statistical Association*, **96**(456), 1348–1360. doi:[10.1198/016214501753382273](https://doi.org/10.1198/016214501753382273).
- Francis B, Dittrich R, Hatzinger R (2010). “Modeling Heterogeneity in Ranked Responses by Nonparametric Maximum Likelihood: How Do Europeans Get Their Scientific Knowledge?” *The Annals of Applied Statistics*, **4**(4), 2181–2202. doi:[10.1214/10-aos366](https://doi.org/10.1214/10-aos366).
- Francis B, Dittrich R, Hatzinger R, Penn R (2002). “Analysing Partial Ranks by Using Smoothed Paired Comparison Methods: An Investigation of Value Orientation in Europe.” *Journal of the Royal Statistical Society C*, **51**(3), 319–336. doi:[10.1111/1467-9876.00271](https://doi.org/10.1111/1467-9876.00271).
- Glenn WA, David HA (1960). “Ties in Paired-Comparison Experiments Using a Modified Thurstone-Mosteller Model.” *Biometrics*, **16**(1), 86–109. doi:[10.2307/2527957](https://doi.org/10.2307/2527957).
- Gneiting T, Raftery A (2007). “Strictly Proper Scoring Rules, Prediction, and Estimation.” *Journal of the American Statistical Association*, **102**(477), 359–376. doi:[10.1198/016214506000001437](https://doi.org/10.1198/016214506000001437).
- Hatzinger R, Dittrich R (2012). “**prefmod**: An R Package for Modeling Preferences Based on Paired Comparisons, Rankings, or Ratings.” *Journal of Statistical Software*, **48**(10), 1–31. doi:[10.18637/jss.v048.i10](https://doi.org/10.18637/jss.v048.i10).
- Luce RD (1959). *Individual Choice Behaviour*. John Wiley & Sons, New York.
- Masarotto G, Varin C (2012). “The Ranking Lasso and Its Application to Sport Tournaments.” *The Annals of Applied Statistics*, **6**(4), 1949–1970. doi:[10.1214/12-aos581](https://doi.org/10.1214/12-aos581).
- Mauerer I, Pössnecker W, Thurner P, Tutz G (2015). “Modeling Electoral Choices in Multi-party Systems with High-Dimensional Data: A Regularized Selection.” *Journal of Choice Modelling*, **16**, 23–42. doi:[10.1016/j.jocm.2015.09.004](https://doi.org/10.1016/j.jocm.2015.09.004).
- Oelker MR (2015). **gvcm.cat**: *Regularized Categorical Effects/Categorical Effect Modifiers/Continuous/Smooth Effects in GLMs*. R package version 1.9, URL <https://CRAN.R-project.org/package=gvcm.cat>.
- Oelker MR, Tutz G (2017). “A Uniform Framework for the Combination of Penalties in Generalized Structured Models.” *Advances in Data Analysis and Classification*, **11**(1), 97–120. doi:[10.1007/s11634-015-0205-y](https://doi.org/10.1007/s11634-015-0205-y).
- Rao PV, Kupper LL (1967). “Ties in Paired-Comparison Experiments: A Generalization of the Bradley-Terry Model.” *Journal of the American Statistical Association*, **62**(317), 194–204. doi:[10.2307/2282923](https://doi.org/10.2307/2282923).

- Rattinger H, Roßteutscher S, Schmitt-Beck R, Weßels B, Wolf C (2014). “Pre-Election Cross Section (GLES 2013).” *GESIS Data Archive, Cologne*, **ZA5700 Data file Version 2.0.0**.
- R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Schauberger G (2019). *BTLLasso: Modelling Heterogeneity in Paired Comparison Data*. R package version 0.1-10, URL <https://CRAN.R-project.org/package=BTLLasso>.
- Schauberger G, Tutz G (2017). “Subject-Specific Modelling of Paired Comparison Data: A Lasso-Type Penalty Approach.” *Statistical Modelling*, **17**(3), 223–243. doi:10.1177/1471082x17693086.
- Springall A (1973). “Response Surface Fitting Using a Generalization of the Bradley-Terry Paired Comparison Model.” *Journal of the Royal Statistical Society C*, **22**(1), 59–68. doi:10.2307/2346303.
- Strobl C, Wickelmaier F, Zeileis A (2011). “Accounting for Individual Differences in Bradley-Terry Models by Means of Recursive Partitioning.” *Journal of Educational and Behavioral Statistics*, **36**(2), 135–153. doi:10.3102/1076998609359791.
- Turner PW, Eymann A (2000). “Policy-Specific Alienation and Indifference in the Calculus of Voting: A Simultaneous Model of Party Choice and Abstention.” *Public Choice*, **102**(1–2), 49–75. doi:10.1023/A:1005046024214.
- Turner H, Firth D (2012). “Bradley-Terry Models in R: The **BradleyTerry2** Package.” *Journal of Statistical Software*, **48**(9), 1–21. doi:10.18637/jss.v048.i09.
- Tutz G (1986). “Bradley-Terry-Luce Models with an Ordered Response.” *Journal of Mathematical Psychology*, **30**(3), 306–316. doi:10.1016/0022-2496(86)90034-9.
- Tutz G (2012). *Regression for Categorical Data*. Cambridge University Press.
- Tutz G, Schauburger G (2015). “Extended Ordered Paired Comparison Models with Application to Football Data from German Bundesliga.” *AStA Advances in Statistical Analysis*, **99**(2), 209–227. doi:10.1007/s10182-014-0237-1.
- Zeileis A, Strobl C, Wickelmaier F, Komboz B, Kopf J (2018). *psychotree: Recursive Partitioning Based on Psychometric Models*. R package version 0.15-2, URL <https://CRAN.R-project.org/package=psychotree>.
- Zermelo E (1929). “Die Berechnung der Turnier-Ergebnisse als ein Maximumsproblem der Wahrscheinlichkeitsrechnung.” *Mathematische Zeitschrift*, **29**(1), 436–460. doi:10.1007/bf01180541.
- Zou H (2006). “The Adaptive Lasso and Its Oracle Properties.” *Journal of the American Statistical Association*, **101**(476), 1418–1429. doi:10.1198/016214506000000735.

**Affiliation:**

Gunther Schaubberger  
Chair of Epidemiology  
Department of Sport and Health Sciences  
Technical University of Munich  
Georg-Brauchle-Ring 56  
80992 München, Germany  
E-mail: [gunther.schauberger@tum.de](mailto:gunther.schauberger@tum.de)

Gerhard Tutz  
Department of Statistics  
Ludwig-Maximilians-Universität München  
Akademiestraße 1  
80799 München, Germany  
E-mail: [tutz@stat.uni-muenchen.de](mailto:tutz@stat.uni-muenchen.de)