Crossover - A search algorithm and GUI for cross-over designs

Kornelius Rohmeyer

March 14, 2023
Chapter 1

Introduction

This package provides more than two hundred cross-over design from literature, a search algorithm to find efficient cross-over designs for various models and a graphical user interface (GUI) to find/generate appropriate designs.

The computationally intensive parts of the package, i.e. the search algorithm, is written using the R packages Rcpp and RcppArmadillo (Eddelbuettel and François [2011] and Eddelbuettel and Sanderson [2013]). The GUI is written in Java and uses package rJava (Urbanek [2013]).

1.1 Installation

Once it is installed, whenever you start R you can load the Crossover package by entering `library(Crossover)` into the R Console. The graphical user interface as shown in figure 1.1 is started with the command `CrossoverGUI()`.

![Figure 1.1: Cross-Over Design GUI.](image-url)
1.2 Overview

The catalogue, collected and compiled by Professor Byron Jones, contains 241 designs from the following literature: Anderson and Preece [2002], Archdeacon et al. [1980], Atkinson [1966], Balaam [1966], Berenblut [1964], Blaisdell Jr and Raghavarao [1980], Davis and Hall [1969], Federer and Atkinson [1964], Fletcher [1987], Iqbal and Jones [1994], Lewis et al. [1988], Cochran et al. [1941], Patterson and Lucas [1962], Pigeon [1985], Prescott [1999], Quenouille [1953], Russell [1991], Lucas [1956], Williams [1949], Prescott [1994], Bate and Jones [2002].

Further 149 designs are constructed from partially balanced incomplete block designs from Clatworthy et al. [1973] and balanced crossover designs.

1.2.1 Design selection in the GUI

The GUI will show appropriate designs from the catalogue according to the number of treatments, periods and the range of sequences the user enters. Further functions from package crossdes [Sailer 2013] are called to create designs for the specified values if possible.

In figure 1.1 you can see the following four checkboxes, that allow you to see only specific subsets:

- **Designs from package archive** The previously noted designs from literature are shown.
- **Designs generated by package crossdes** Activating this option will result in short delays when displaying the catalogue, since the crossdes algorithms are called.
- **Designs manually entered** All designs entered on tab "Input own design" are shown.
- **Designs from previous search runs** All designs from previous search runs are shown.

The column av.e.trt.pair shows the average efficiency of pairwise treatment parameter differences.

To define these efficiencies we compare a design $D$ with an "ideal design" $D_I$ without period, subject or carry-over effects, where each treatment $i \in \{1, \ldots, v\}$ occurs $n_i$ times in $D$ and $D_I$. For $D_I$ the variance of the treatment parameter difference estimate $\hat{\tau}_i - \hat{\tau}_j$ for treatment $i$ and $j$ is $1/n_i + 1/n_j$. We define the efficiency for a design $D$ as the quotient of the variances under $D_I$ and $D$, i.e.

$$\frac{\text{Var}_{D_I}(\hat{\tau}_i - \hat{\tau}_j)}{\text{Var}_D(\hat{\tau}_i - \hat{\tau}_j)} = \frac{1/n_i + 1/n_j}{\text{Var}_D(\hat{\tau}_i - \hat{\tau}_j)}.$$ 

The tab "Input own design" provides you with the possibility to analyse and save your own designs easily or to use them as starting points for the search algorithm.

The drop-down menu for model let you specify which model you are interested. These models are described in detail in chapter 2. In case of the placebo or proportionality model you can specify further model parameters (namely the number of placebos and the proportionality parameter, respectively).

---

1A table referencing all available designs and the respective number of treatments, periods and sequences is available by calling buildSummaryTable().
1.2.2 Designs in R

Our crossover designs in R are numeric matrices, where the elements represent the treatments, the rows represent the periods and the columns the subjects.

A data frame referencing all available designs and the respective number of treatments, periods and sequences is available by calling the `buildSummaryTable()` function.

Designs referenced in this table can be accessed via the `getDesign()` function. For example the Williams design for three treatments is represented by a $3 \times 6$-matrix and assigns each of the three treatments once to each of the six subjects:

```r
getDesign("williams3t")
```

```
## s1 s2 s3 s4 s5 s6
## p1 1 2 3 3 1 2
## p2 2 3 1 2 3 1
## p3 3 1 2 1 2 3
## attr(, "reference")
## attr(, "signature")
## [1] "p = 3, n = 6, t = 3"
## attr(, "title")
## [1] "WILLIAMS DESIGN THREE TREATMENTS"
```

Each treatment occurs six times, two times in each period and each treatment follows each other treatment exactly two times.

If we are interested in the variance of the treatment parameter difference estimates, we can use the function `general.carryover`:

```r
design <- getDesign("williams3t")
general.carryover(design, model=1)
```

```
## $Var.trt.pair
## [,1] [,2] [,3]
## [1,] 0.0000 0.4167 0.4167
## [2,] 0.4167 0.0000 0.4167
## [3,] 0.4167 0.4167 0.0000
##
## $Var.car.pair
## [,1] [,2] [,3]
## [1,] 0.00 0.75 0.75
## [2,] 0.75 0.00 0.75
## [3,] 0.75 0.75 0.00
##
## $model
## [1] 1
```

We see that the Williams design is a balanced design.

The following nine models which are discussed in chapter 2 are implemented:

```r
## 1: "Standard additive model"
## 2: "Self-adjacency model"
## 3: "Proportionality model"
## 4: "Placebo model"
## 5: "No carry-over into self model"
## 6: "Treatment decay model"
## 7: "Full set of interactions"
## 8: "Second-order carry-over effects"
## 9: "No carry-over effects"
```
1.2.3 Algorithm Search

In figure 1.2 the preliminary graphical interface for the search algorithm is shown with the following options:

- The covariance pattern can be "Independence", "Autoregressive Error" or "Autoregressive Error". Except for "Independence" a covariance pattern coefficient $\rho$ has to be specified.

- Specify the exact number of sequences. (The number of treatments and periods is already specified in the top panel of the GUI.)

- Optionally specify the exact number of treatment assignments. The GUI default is to let the algorithm figure out good/optimal assignments. But depending on further information (information from theoretical results or treatments more important than the others etc.) the number of treatment assignments can be specified.

- You can specify that the design should be constructed in a way that in each sequence/period a treatment occur as evenly as possible. This restriction will normally decrease the efficiency of the algorithm.

- The GUI default is an all-pair comparison of all treatments with equal weights. Change the contrast weights accordingly if you are interested in other contrasts or different weights.

- Pressing the "Compute Design" button will start the search algorithm described in section 3. After a few seconds the result will be shown in the previous empty text area on the right.

---

2 Different weights of treatment importance should be specified as weighted contrasts. See item contrasts.
Chapter 2

Models

The package and GUI support the following eight models.

2.1 Standard additive model

\[ Y_{ijk} = \mu + \pi_j + \tau_{d[i,j]} + \lambda_{d[i,j-1]} + s_{ik} + e_{ijk} \]

with

- \( \mu \) intercept,
- \( \pi_j \) period effect for period \( j \),
- \( \tau_{d[i,j]} \) direct treatment effect for treatment \( d[i,j] \) in period \( j \) of sequence \( i \),
- \( \lambda_{d[i,j-1]} \) first-order carry-over effect (0 for \( j-1=0 \)),
- \( s_{ik} \) \( k \)th subject effect on sequence \( i \),
- \( e_{ijk} \) random error with zero mean and variance \( \sigma^2 \).

which we can write as

\[ E(Y) = \mu + X (\tilde{\pi}) + Z (\tilde{\tau}) \]

with \( X \) and \( Z \) called the treatment and block design matrices, respectively.

We call \( H \) a link matrix if \( X = X_r H \) were \( X_r \) is the design matrix for the row-column design. The rows of the link matrix specify all possible parameter combinations. Therefore \( H \) has as many columns as there are parameters and in the row-column-setting \( i \) parameter \( j \) is included \( h_{ij} \) times.

```r
# Design:

design <- rbind(c(3,2,1),
                 c(2,1,3),
                 c(1,2,3),
                 c(3,2,1))
design
## [,1] [,2] [,3]
## [1,] 3 2 1
## [2,] 2 1 3
## [3,] 1 2 3
## [4,] 3 2 1
```

\(^1\text{cf. Jones and Kenward [2003], page 8}\)
v <- 3 # number of treatments

# Link matrix:
H <- Crossover:::linkMatrix(model="Standard additive model", v)
H

## [1,] 1 0 0 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 0 0 1 0 0 0
## [4,] 1 0 0 1 0 0
## [5,] 0 1 0 1 0 0
## [6,] 0 0 1 1 0 0
## [7,] 1 0 0 0 1 0
## [8,] 0 1 0 0 1 0
## [9,] 0 0 1 0 1 0
## [10,] 1 0 0 0 0 1
## [11,] 0 1 0 0 0 1
## [12,] 0 0 1 0 0 1

# Row-Column-Design: (cf. John et al. 2004, Table II and page 2649f.)
rcDesign <- Crossover:::rcd(design, v=v, model=1)
rcDesign

## [,1] [,2] [,3]
## [1,] 3 2 1
## [2,] 11 7 6
## [3,] 7 5 12
## [4,] 6 8 10

# Design Matrix of Row-Column Design:
Xr <- Crossover:::rcdMatrix(rcDesign, v, model=1)
Xr

## [1,] 0 0 1 0 0 0 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0 0 0 0
## [3,] 1 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 1
## [5,] 0 0 0 0 0 0 1 0 0 0 0 0
## [6,] 0 0 0 0 0 1 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 1 0 0 0 0 0
## [8,] 0 0 0 0 1 0 0 0 0 0 0 0
## [9,] 0 0 0 0 0 0 0 0 0 0 0 1
## [10,] 0 0 0 0 0 0 0 0 0 0 0 0
## [11,] 0 0 0 0 0 0 0 0 0 1 0 0
## [12,] 0 0 0 0 0 0 0 0 0 0 1 0

# Design Matrix of Cross-Over Design:
X <- Xr %*% H
X

## [1,] 0 0 1 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 1 0 0 0 0 0
## [4,] 0 1 0 0 1 0
## [5,] 1 0 0 0 1 0
## [6,] 0 1 0 1 0 0
## [7,] 1 0 0 0 0 1
## [8,] 0 1 0 1 0 1
## [9,] 0 1 0 0 1 0
## [10,] 1 0 0 0 1 0
## [11,] 0 1 0 0 1 0
## [12,] 1 0 0 0 0 1
2.2 Full set of interactions

This model has further interaction parameters $\gamma_{ij}$ and a design matrix $X$ with $2 \times v + v^2$ columns:

$$E(Y) = \mu + X \left( \begin{array}{c} \gamma_1 \\ \gamma_2 \end{array} \right) + Z \bar{\gamma}$$

```r
H <- Crossover::linkMatrix(model="Full set of interactions", v)

# Design Matrix of Cross-over Design:
X <- Xr %*% H
```

```r
## [1,] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0
## [5,] 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0
## [6,] 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0
## [7,] 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0
## [8,] 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0
## [9,] 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0
## [10,] 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
## [11,] 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0
## [12,] 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0

# Design Matrix of Cross-over Design:
X <- Xr %*% H

## [1,] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [5,] 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0
## [6,] 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
## [7,] 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [9,] 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0
## [10,] 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
## [11,] 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [12,] 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0

## [1,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [9,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [10,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [11,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [12,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2.3 Self-adjacency model

In the self-adjacency model $\lambda_i$ is replaced by $\varphi_i$ in case of carry-over into itself. The case $\varphi = 0$ represents the no carry-over into self model.

$$E(Y) = \mu + X(\tilde{\lambda}) + Z(\pi)$$

H <- Crossover:::linkMatrix(model="Self-adjacency model", v)
H
## [1,] 1 0 0 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0
## [3,] 0 0 1 0 0 0 0 0 0
## [4,] 1 0 0 0 0 0 1 0 0
## [5,] 0 1 0 1 0 0 0 0 0
## [6,] 0 0 1 1 0 0 0 0 0
## [7,] 1 0 0 0 1 0 0 0 0
## [8,] 0 1 0 0 0 0 0 1 0
## [9,] 0 0 1 0 1 0 0 0 0

# Design Matrix of Cross-Over Design:
X <- Xr %*% H
X
## [1,] 0 0 1 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0
## [3,] 1 0 0 0 0 0 0 0 0
## [4,] 0 1 0 0 0 1 0 0 0
## [5,] 1 0 0 0 1 0 0 0 0
## [6,] 0 0 1 1 0 0 0 0 0
## [7,] 1 0 0 0 1 0 0 0 0
## [8,] 0 1 0 1 0 0 0 0 0
## [9,] 0 0 1 1 0 0 0 0 0
## [10,] 0 0 1 0 0 0 0 0 1
## [11,] 0 1 0 0 0 0 0 1 0
## [12,] 1 0 0 0 0 1 0 0 0

2.4 Placebo model

In the placebo model there are no carry-over effects for the placebo treatment(s).

$$E(Y) = \mu + X(\tilde{\lambda}) + Z(\pi)$$

# link matrix:
H <- Crossover:::linkMatrix(model="Placebo model", v, placebos=1)
H
## [1,] 1 0 0 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0
## [3,] 0 0 1 0 0 0 0 0 0
## [4,] 1 0 0 0 0 0 0 0 0
## [5,] 0 1 0 0 0 0 0 0 0
## [6,] 0 0 1 0 0 0 0 0 0
## [7,] 1 0 0 0 1 0 0 0 0
## [8,] 0 1 0 1 0 0 0 0 0
## [9,] 0 0 1 0 0 0 0 0 0
## [10,] 0 0 1 0 0 0 0 0 0
## [11,] 0 1 0 0 0 0 0 1 0
## [12,] 1 0 0 0 0 1 0 0 0
# Design Matrix of Cross-Over Design:

```r
X <- Xr %*% H
X
```

##

<table>
<thead>
<tr>
<th></th>
<th>[1,]</th>
<th>[2,]</th>
<th>[3,]</th>
<th>[4,]</th>
<th>[5,]</th>
<th>[6,]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0</td>
<td>0 1 0</td>
<td>0 0 1</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0</td>
<td>0 0 1</td>
<td>0 1 0</td>
<td>0 0 1</td>
<td>0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>3</td>
<td>1 0 0</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 1 0</td>
<td>0 0 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>4</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 0 1</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>5</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 0 1</td>
<td>0 1 0</td>
</tr>
<tr>
<td>6</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

2.5 No carry-over into self model

This model differs from the standard additive model in the assumption that in the no carry-over into self model no carry-over effect occurs if current and previous treatment are the same.

\[
E(Y) = \mu + X(\tilde{\lambda}) + Z(\tilde{\pi})
\]

```r
H <- Crossover:::linkMatrix(model="No carry-over into self model", r)
H
```

##

<table>
<thead>
<tr>
<th></th>
<th>[1,]</th>
<th>[2,]</th>
<th>[3,]</th>
<th>[4,]</th>
<th>[5,]</th>
<th>[6,]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>3</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>6</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>7</td>
<td>1 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>8</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>9</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>10</td>
<td>1 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>11</td>
<td>0 1 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>12</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

# Design Matrix of Cross-Over Design:

```r
X <- Xr %*% H
X
```

##

<table>
<thead>
<tr>
<th></th>
<th>[1,]</th>
<th>[2,]</th>
<th>[3,]</th>
<th>[4,]</th>
<th>[5,]</th>
<th>[6,]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>3</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>6</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>7</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>8</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>9</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>10</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>11</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>12</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>
2.6 Treatment decay model

In contrast to the no carry-over into self model in the treatment decay model we assume there are only (negative) carry-over effects if the current and previous treatment are the same.

\[ E(Y) = \mu + X(\bar{\tau}) + Z(\bar{\tau}) \]

```r
H <- Crossover:::linkMatrix(model="Treatment decay model", r)
H
```

## [1,] 1 0 0 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 0 0 1 0 0 0
## [4,] 1 0 0 -1 0 0
## [5,] 0 1 0 0 0 0
## [6,] 0 0 1 0 0 0
## [7,] 1 0 0 0 0 0
## [8,] 0 1 0 0 -1 0
## [9,] 0 0 1 0 0 0
## [10,] 1 0 0 0 0 0
## [11,] 0 1 0 0 0 0
## [12,] 0 0 1 0 0 -1

# Design Matrix of Crossover Design:
X <- Xr %*% H
X
```

## [1,] 0 0 1 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 1 0 0 0 0 0
## [4,] 0 1 0 0 0 0
## [5,] 1 0 0 0 0 0
## [6,] 0 0 1 0 0 0
## [7,] 1 0 0 0 0 0
## [8,] 0 1 0 0 0 0
## [9,] 0 0 1 0 0 -1
## [10,] 0 0 1 0 0 0
## [11,] 0 1 0 0 -1 0
## [12,] 1 0 0 0 0 0

2.7 Proportionality model

For the proportionality model we have no separate carry-over effects, but assume for period \( i > 1 \) an additive proportion \( p \) of the effect \( \tau_{d(i-1,j)} \) from the previous treatment \( d(i-1,j) \). This model is non-linear in \( p \), therefore we assume \( p \) is known.

\[ E(Y) = \mu + X_p\tau + Z(\bar{\tau}) \]

```r
H <- Crossover:::linkMatrix(model="Proportionality model", r)
H
```

## [,1] [,2] [,3]
## [1,] 1.0 0.0 0.0
## [2,] 0.0 1.0 0.0
## [3,] 0.0 0.0 1.0
## [4,] 1.5 0.0 0.0
## [5,] 0.5 1.0 0.0
## [6,] 0.5 0.0 1.0
## Design Matrix of Cross-Over Design:

```r
X <- Xr %*% H
X
```

```text
[1,] 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
[2,] 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
[3,] 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
[4,] 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
[5,] 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
[6,] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
[7,] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
[8,] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1
[9,] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
[10,] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
[11,] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
[12,] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
```

2.8 Second-order carry-over effects

In the model with second-order carry-over effects we do have another vector $\lambda_2$ of carry-over effects:

$$E(Y) = \mu + X \left( \frac{\lambda_1}{\lambda_2} \right) + Z \left( \frac{\pi}{s} \right)$$

Therefore the link matrix has $3 \times v$ columns.

```r
# Link matrix:
H <- Crossover:::linkMatrix(model="Second-order carry-over effects", v)
H
```

```text
[1,] 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
[2,] 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
[3,] 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
[4,] 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
[5,] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
[6,] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
[7,] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1
[8,] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
[9,] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
[10,] 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
[11,] 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
[12,] 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0
[13,] 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
[14,] 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0
[15,] 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
[16,] 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
[17,] 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0
[18,] 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0
[19,] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0
[20,] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0
[21,] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
[22,] 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
[23,] 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
[24,] 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
[25,] 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
[26,] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
```
# Row-Column Design: (cf. John et al. 2004, Table II and page 2649f.)

```r
cdDesign <- Crossover:::rcd(design, v=v, model=8)
cdDesign
```

## [,1] [,2] [,3]
## [1,] 3 2 1
## [2,] 11 7 6
## [3,] 34 23 21
## [4,] 24 17 37

# Design Matrix of Row-Column Design:

```r
Xr <- Crossover:::rcdMatrix(cdDesign, v, model=8)
Xr
```

## [1,] 0 0 1 0 0 0 0 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0 0 0 0 0
## [3,] 1 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 1 0 0 0 0 0 0
## [5,] 0 0 0 0 0 1 0 0 0 0 0 0 0
## [6,] 0 0 0 0 1 0 0 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [9,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [10,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [11,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [12,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [9,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [10,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [11,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [12,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [9,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [10,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [11,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [12,] 0 0 0 0 0 0 0 0 0 0 0 0 0
## [,38] [,39]
## [1,] 0 0
## [2,] 0 0
Design Matrix of Cross-Over Design:

\[
X \leftarrow X_r \times H
\]

```
[[1,] 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1
[[2,] 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1
[[3,] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[[4,] 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[[5,] 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[[6,] 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[[7,] 1 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1
[[8,] 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 1
[[9,] 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1
[[10,] 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1
[[11,] 0 1 0 1 0 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1
[[12,] 1 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 1 1 0 0 0 1
```
```
Chapter 3

Search strategy

Monte-Carlo search for first candidates, followed by parallel hill climbing search, extended by long jumps. In the following graphic we see 100,000 design evaluations (that take less than 7 seconds on my computer), consisting out of 20 hill climbing search runs from 20 different designs as starting point which were selected by a Monte-Carlo search. We can see that 10 out of the 20 search runs were not able to achieve the efficiency the other 10 achieved. We can see, that there were most likely at least four local maxima the hill climbing search found and got stuck:

```r
set.seed(42)
x <- searchCrossOverDesign(n=9, p=5, v=4, model=4)
plot(x)
```
plot(x, type="b")
**Definition 3.0.1 (Efficiency)**. If the difference of two treatments, replicated $r_1$ and $r_2$ times, is estimated in an ideal design with residual variance $\sigma^2$, the variance of the estimated difference $\bar{y}_A - \bar{y}_B$ is

$$V_I = \left( \frac{1}{r_1} + \frac{1}{r_2} \right) \sigma^2.$$  

In a crossover design the variance of a parameter estimates for $A$ and $B$ are given by the corresponding elements $\sigma^2 x_A$ and $\sigma^2 x_B$ of $\sigma^2 (X^T X)^{-1}$

with $X$ the crossover design matrix. The variance of the difference is $\sigma^2 (x_A + x_B - 2 \star x_{AB})$ where $x_{AB}$ is the element of $X^T X$ specifying the covariance of the two treatment parameter estimates. The efficiency is

$$E := \frac{V_I}{V_C} = \frac{r_1 + r_2}{r_1 r_2 (x_A + x_B - 2 \star x_{AB})} \in [0, 1].$$

If we have a model with $n$ parameters and a contrast matrix $C \in \mathbb{R}^{n \times n}$, the sum of the variances of the
linear combinations given by $C$ is
\[ \text{trace}(CH^T A_r HCT). \]

**Example for two treatments**

We compare our results with results from the algorithm of John et al. [2004] that are presented in Jones and Kenward [2003] table 4.35-4.38, p. 202ff for $s=6$ sequences, $p=4$ periods and $v=2$ treatments under the constraint that each treatment should occur 12 times.

The following shows all designs that were found. For some models the algorithm always returned only one design, but for some 2 or even 3 equally good designs were found.

Here are all the designs as found by the above code. Actually there are only 11 different designs, since the following designs were already found for other models: 7, 8, 9, 12, 13.

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 2 & 2 & 1 & 1 & 2 \\
2 & 2 & 2 & 1 & 1 & 1 \\
2 & 1 & 1 & 2 & 2 & 1 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 2 & 1 & 1 & 1 \\
\end{array}
\]

**Design 1**

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
\end{array}
\]

**Design 2**

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
2 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 1 & 2 & 1 & 1 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 2 & 1 & 2 & 1 & 1 \\
1 & 2 & 1 & 2 & 1 & 1 \\
\end{array}
\]

**Design 3**

\[
\begin{array}{cccccc}
1 & 2 & 1 & 2 & 1 & 1 \\
1 & 2 & 1 & 2 & 1 & 2 \\
2 & 1 & 2 & 1 & 2 & 1 \\
2 & 1 & 2 & 1 & 2 & 1 \\
1 & 2 & 1 & 2 & 1 & 1 \\
1 & 2 & 1 & 2 & 1 & 2 \\
1 & 2 & 1 & 2 & 1 & 1 \\
\end{array}
\]

**Design 4**

\[
\begin{array}{cccccc}
1 & 1 & 2 & 1 & 2 & 2 \\
2 & 2 & 2 & 1 & 1 & 1 \\
1 & 1 & 2 & 1 & 2 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
2 & 2 & 1 & 1 & 2 & 1 \\
1 & 1 & 2 & 1 & 2 & 2 \\
1 & 1 & 2 & 1 & 2 & 2 \\
\end{array}
\]

**Design 5**

\[
\begin{array}{cccccc}
1 & 2 & 2 & 1 & 1 & 2 \\
2 & 1 & 2 & 1 & 2 & 1 \\
1 & 2 & 1 & 2 & 1 & 2 \\
2 & 1 & 2 & 1 & 2 & 1 \\
1 & 2 & 1 & 2 & 1 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
\end{array}
\]

**Design 6**

\[
\begin{array}{cccccc}
1 & 1 & 2 & 1 & 2 & 2 \\
1 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 2 & 1 & 1 & 1 \\
1 & 1 & 2 & 1 & 2 & 2 \\
1 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 2 & 1 & 1 & 1 \\
\end{array}
\]

**Design 7**

\[
\begin{array}{cccccc}
1 & 1 & 2 & 1 & 2 & 2 \\
2 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 1 & 1 & 2 & 1 \\
2 & 2 & 1 & 1 & 2 & 1 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
\end{array}
\]

**Design 8**

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
\end{array}
\]

**Design 9**

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
\end{array}
\]

**Design 10**

\[
\begin{array}{cccccc}
1 & 1 & 2 & 1 & 2 & 2 \\
2 & 2 & 2 & 1 & 1 & 1 \\
1 & 1 & 1 & 2 & 2 & 2 \\
2 & 2 & 1 & 1 & 2 & 1 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
\end{array}
\]

**Design 11**

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
\end{array}
\]

**Design 12**

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 1 & 1 & 2 & 2 & 2 \\
\end{array}
\]

**Design 13**

\[
\begin{array}{cccccc}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 2 & 2 & 1 & 1 & 2 \\
2 & 1 & 1 & 2 & 2 & 1 \\
1 & 2 & 1 & 2 & 1 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
\end{array}
\]

**Design 14**

\[
\begin{array}{cccccc}
1 & 2 & 1 & 2 & 1 & 2 \\
2 & 1 & 2 & 1 & 2 & 1 \\
1 & 2 & 1 & 2 & 1 & 2 \\
2 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 2 & 1 & 1 & 1 \\
2 & 2 & 2 & 1 & 1 & 1 \\
\end{array}
\]

**Design 15**

\[
\begin{array}{cccccc}
1 & 2 & 1 & 2 & 1 & 2 \\
2 & 2 & 1 & 1 & 2 & 1 \\
1 & 1 & 1 & 2 & 2 & 2 \\
2 & 2 & 1 & 1 & 2 & 1 \\
1 & 1 & 1 & 2 & 2 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
1 & 2 & 1 & 2 & 1 & 2 \\
\end{array}
\]

**Design 16**
<table>
<thead>
<tr>
<th>Design</th>
<th>Additive</th>
<th>Self-adjacency</th>
<th>Proportional</th>
<th>Placebo</th>
<th>No into self</th>
<th>Decay</th>
<th>Interaction</th>
<th>2nd-order carry-over</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.042</td>
<td>0.231</td>
<td>0.038</td>
<td>0.042</td>
<td>0.074</td>
<td>0.067</td>
<td>0.231</td>
<td>0.111</td>
</tr>
<tr>
<td>2</td>
<td>0.042</td>
<td>0.231</td>
<td>0.033</td>
<td>0.042</td>
<td>0.092</td>
<td>0.062</td>
<td>0.236</td>
<td>0.050</td>
</tr>
<tr>
<td>3</td>
<td>0.068</td>
<td>0.236</td>
<td>0.066</td>
<td>0.068</td>
<td>0.134</td>
<td>0.053</td>
<td>0.223</td>
<td>0.090</td>
</tr>
<tr>
<td>4</td>
<td>0.068</td>
<td>0.233</td>
<td>0.066</td>
<td>0.068</td>
<td>0.134</td>
<td>0.053</td>
<td>0.223</td>
<td>0.088</td>
</tr>
<tr>
<td>5</td>
<td>0.046</td>
<td>0.250</td>
<td>0.019</td>
<td>0.046</td>
<td>0.062</td>
<td>0.083</td>
<td>Not estimable</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.042</td>
<td>0.231</td>
<td>0.033</td>
<td>0.042</td>
<td>0.067</td>
<td>0.074</td>
<td>0.231</td>
<td>0.168</td>
</tr>
<tr>
<td>7</td>
<td>0.042</td>
<td>0.231</td>
<td>0.038</td>
<td>0.042</td>
<td>0.082</td>
<td>0.062</td>
<td>0.231</td>
<td>0.074</td>
</tr>
<tr>
<td>8</td>
<td>0.049</td>
<td>0.236</td>
<td>0.032</td>
<td>0.049</td>
<td>0.061</td>
<td>0.083</td>
<td>0.232</td>
<td>0.074</td>
</tr>
<tr>
<td>9</td>
<td>0.229</td>
<td>0.229</td>
<td>0.099</td>
<td>0.229</td>
<td>0.229</td>
<td>0.229</td>
<td>0.250</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.068</td>
<td>0.233</td>
<td>0.066</td>
<td>0.068</td>
<td>0.134</td>
<td>0.053</td>
<td>0.223</td>
<td>0.088</td>
</tr>
<tr>
<td>11</td>
<td>0.047</td>
<td>0.250</td>
<td>0.043</td>
<td>0.047</td>
<td>0.090</td>
<td>0.063</td>
<td>0.250</td>
<td>0.047</td>
</tr>
<tr>
<td>12</td>
<td>0.05</td>
<td>0.234</td>
<td>0.090</td>
<td>0.120</td>
<td>0.203</td>
<td>0.047</td>
<td>0.234</td>
<td>0.173</td>
</tr>
<tr>
<td>13</td>
<td>0.068</td>
<td>0.233</td>
<td>0.066</td>
<td>0.068</td>
<td>0.134</td>
<td>0.053</td>
<td>0.223</td>
<td>0.088</td>
</tr>
<tr>
<td>14</td>
<td>0.12</td>
<td>0.234</td>
<td>0.090</td>
<td>0.120</td>
<td>0.203</td>
<td>0.047</td>
<td>0.234</td>
<td>0.135</td>
</tr>
</tbody>
</table>

Note that for the full interaction model and the self-adjacency model we get the same results, because they are essentially the same models with

\[
\lambda^{SA}_1 = \lambda^{FI}_1 + \gamma^{FI}_{12}, \]
\[
\lambda^{SA}_2 = \lambda^{FI}_2 + \gamma^{FI}_{21}, \]
\[
\varphi^{SA}_1 = \lambda^{FI}_1 + \gamma^{FI}_{11}, \text{ and} \]
\[
\varphi^{SA}_2 = \lambda^{FI}_2 + \gamma^{FI}_{22}. \]

For the treatment decay model the algorithm provided us with a design where the treatment decay effect never occurs. Let’s give a little bit of weight to this treatment decay effect:

As we can see we find the design from the book, when we give weights 100:1 for treatment / carry-over effects in the treatment decay model. (As seen in the first result section a zero weight for the carry-over effects results in a model with alternating treatments, so that there never occurs a treatment decay effect.)

**Summary**

### 3.1 Random Subject Effects Model

See Jones and Kenward [2003], 5.3, page 213ff. The model stays the same

\[ Y_{ijk} = \mu + \pi_j + \tau_{d[i,j]} + \lambda_{d[i,j-1]} + s_{ik} + e_{ijk}, \]

but we also assume that the subject effects follow a normal distribution: \( s_{ik} \sim N(0, \sigma_s^2) \).

In matrix notation we have

\[ Y = X\beta + Z\gamma + \varepsilon \]

with \( X \) and \( Z \) the fixed and random effects design matrices\(^1\) \( \varepsilon \sim N(0, \Sigma) \) and \( \gamma \sim N(0, D) \). Then\(^2\)

\[ \text{Var}(Y) = ZDZ^T + \Sigma. \]

For known \( V := \text{Var}(Y) = ZDZ^T + \Sigma \) the MLE and BLUE is given by

\[ \hat{\beta} = (X^tV^{-1}X)^{-1}X^tV^{-1}Y. \]

---

\(^1\)Note that \( X \) and \( Z \) are different from the ...

\(^2\)See for example [Lee et al., 2006, chapter 5].
Appendix A

Appendix

The appendices are all work in progress.

A.1 Appendix - Variance Balance / Partial Balance

Definition A.1.1 (Variance Balance). A design is variance balanced if $\text{Var}[\hat{\tau}_i - \hat{\tau}_j]$ is constant for all pairs of the $n$ treatment effects $\tau_i, \tau_j, 1 \leq i, j \leq n, i \neq j$.

Definition A.1.2 (Partially balanced incomplete block design with two associate classes). A partially balanced incomplete block design with two associate classes (or short PBIBD(2)) is a design that fulfils the following (see Clatworthy et al. [1973]):

- There are $b$ blocks of $k$ units each.
- There are $v$ treatments and $v > k$, so that not all treatments can occur in a block. Each treatment occurs once in each block and is assigned $r$ times.
- Let $X$ be the set of treatments. Then there is a partition of $X \times X$ into three relations, with one relation the identity and the other two relations $R_1$ and $R_2$ fulfilling the following properties:
  - A treatment $y$ is called an $i$th associates, $i \in \{1, 2\}$, if and only if $(x, y) \in R_i$. Each treatment $x$ has $n_i$ $i$th associates.
  - For all $i, j, k \in \{1, 2\}$ there is a constant numbers $p^i_{jk}$, so that for all $(x, y) \in R_i$ the number of treatments $z$ with $(x, z) \in R_j$ and $(z, y) \in R_k$ is $p^i_{jk}$.
  - The association scheme is commutative, i.e. $p^i_{jk} = p^j_{ki}$.
- There are two numbers $\lambda_1, \lambda_2$ so that $(x, y) \in R_i, i \in \{1, 2\}$, implies that treatment $x$ and $y$ occur together in exactly $\lambda_i$ blocks.

A.2 Appendix - Matrix Algebra / Linear Models

A.2.1 Generalized inverse

Definition A.2.1 (Moore-Penrose pseudoinverse). For $A \in \mathbb{R}^{m \times n}$ a matrix $A^+$ is called Moore-Penrose pseudoinverse iff

1. $AA^+A = A$, 

20
2. \( A^+AA^+ = A^+ \),
3. \((AA^+)^T = AA^+\) and
4. \((A^+A)^T = A^+A\).

The Moore-Penrose pseudoinverse exists for every real matrix and is unique.

**Theorem A.2.2** (Block matrix pseudoinverse). For a block matrix

\[
X := [A \ B]
\]

we have

\[
(X^TX)^+ = \left( \begin{bmatrix} A^T & B^T \end{bmatrix} \begin{bmatrix} A \ B \end{bmatrix} \right)^+ = \left( \begin{bmatrix} A^T A & A^T B \\ B^T A & B^T B \end{bmatrix} \right)^+ = \begin{bmatrix} (A^T P_A^T A)^+ & -(A^T P_A^T A)^+ A^T B(B^T B)^+ \\ -(B^T P_A^T B)^+ B^T A(A^T A)^+ & (B^T P_A^T B)^+ \end{bmatrix}
\]

with the orthogonal projection matrices

\[
P_B^T = I - B(B^T B)^+ B^T, \quad P_A^T = I - A(A^T A)^+ A^T.
\]


**Theorem A.2.3.** Let \( Ax = b \) a linear equation system in \( n \) variables and \( A^+ \) the Moore-Penrose pseudoinverse of \( A \). Solutions exist if and only if \( AA^+ b = b \) and are given in this case by

\[
\{ A^+ b + (I - A^+A)y \mid y \in \mathbb{R}^n \}.
\]

**Proof.** If \( AA^+ b = b \) we have for every \( A^+ b + (I - A^+A)y \) with \( y \in \mathbb{R}^n \):

\[
A(A^+ b + (I - A^+A)y) = b + Ay - AA^+Ay = b + Ay - Ay = b
\]

... \( \square \)

**Theorem A.2.4.** Let \( y = X\beta + \varepsilon \) be a linear model with \( \varepsilon \sim N(0, \sigma^2I) \). Then \( t^T \beta \) is estimable iff \( t^T (X^TX)^+X^TX = t^T \).

### A.2.2 Fisher information

For a linear regression model with design matrix \( X \) and variance \( \sigma^2 \) the information matrix is given by

\[
\frac{1}{\sigma^2}X^TX.
\]

### A.2.3 Linear Mixed Models

In the linear mixed model the response \( Y \) is given by

\[
Y = X\beta + Z\gamma + \varepsilon
\]

with \( X \) and \( Z \) the fixed and random effects design matrices, \( \varepsilon \sim N(0, \Sigma) \) and \( \gamma \sim N(0, D) \). Then

\[
\text{Var}(Y) = ZDZ^T + \Sigma.
\]

For known \( V := \text{Var}(Y) = ZDZ^T + \Sigma \) the MLE and BLUE is given by

\[
\hat{\beta} = (X^TV^{-1}X)^{-1}X^TV^{-1}Y.
\]

\[1\text{See for example [Lee et al., 2006, chapter 5].}\]
A.3 Appendix - Internals

A.3.1 Naming conventions

Most functions exist as R and C/C++ variant, with the following naming conventions:

We have functions in R and on the C/C++ level with the same name (in our following example rcd) that should be used in R and C/C++ respectively. The R function rcd calls the C function rcd2R which handles the conversion from and to the R structures the SEXP are pointing to. Apart from that we often provide an R function with a suffix _R, i.e. in our example rcd_R, which is a pure R function and reimplements the same functionality for testing purposes.

\[
\text{SEXP createRCDesign2R( SEXP designS, SEXP vS, SEXP modelS) } \{ \ldots \}
\]
\[
\text{arma::mat createRCDesign(arma::mat design, int v, int model) } \{ \ldots \}
\]

\[
\text{createRCDesign <- function (X, v, model) } \{
\text{ return (.Call("createRCDesign2R", X, v, model, PACKAGE = "Crossover")) }
\}
\]
\[
\text{createRCDesign_R <- function (X, v, model) } \{
\text{ # Same functionality written in R }
\}
\]

A variable \(v\) means always the original number of treatments. A variable \(vv\) stands for the maximum number of different settings\(^2\), either \(v + v^2\) or for the second-order carry-over effects \(v + v^2 + v^3\).

\(^2\)For the no carry-over into self or placebo model there are actually less settings.
Index

balanced
  variance, 20
block design matrix, 6

efficiency, 17

Fisher information, 21

information matrix, 21

Linear Mixed Models, 21

link matrix, 6

model
  no carry-over into self, 10
  placebo model, 9
  proportionality model, 11
  self-adjacency model, 9
  standard additive, 8
  treatment decay, 11
  with full set of interactions, 8
  with second-order carry-over effects, 12
Moore-Penrose pseudoinverse, 20

partially balanced incomplete block design, 20
PBIB(2), 20

second-order carry-over effects, 12
self-adjacency model, 9

treatment design matrix, 6

variance balance, 20
Bibliography


Table of Symbols

Sets
\( \mathbb{R} \)  set of real numbers
\( \mathbb{N}_0 \) set of natural numbers (including 0)
\( \mathcal{P}(X) \) power set of set \( X \), i.e. the set of all subsets of \( X \)

Variables
\( v \)  number of treatments
\( p \)  number of periods
\( s \)  number of sequences
\( \mu \)  intercept
\( \pi_j \)  period effect for period \( j \)
\( \tau_{d[i,j]} \) direct treatment effect for treatment \( d[i,j] \) in period \( j \) of sequence \( i \)
\( \lambda_{d[i,j-1]} \) first-order carry-over effect (0 for \( j - 1 = 0 \))
\( s_{ik} \)  \( k \)th subject effect on sequence \( i \)
\( \epsilon_{ijk} \) random error with zero mean and variance \( \sigma^2 \)

Functions
\( X' \) transpose of matrix \( X \)
\( X^+ \) Moore-Penrose pseudoinverse of \( X \)
\( \langle \cdot, \cdot \rangle \) standard direct product \( \langle x, y \rangle = \sum_{j=1}^{n} x_j \cdot y_j \) for \( x, y \in \mathbb{R}^n \)
\( \text{id}_X \) identity on \( X \), i.e. \( \text{id}_X : X \to X, x \mapsto x \)

Other Symbols
\( \mathcal{N}(\mu, \sigma^2) \) Normal distribution with mean \( \mu \) and variance \( \sigma^2 \).
\( \mathcal{N}(\mu, \Sigma) \) Multivariate normal distribution with mean \( \mu \) and covariance matrix \( \Sigma \).