Drawing diagrams useful for latent scales

Michael Dewey

April 6, 2020

1 Introduction

This document describes `latdiag` a routine for displaying and checking response patterns in a way useful in latent trait modelling.

I assume that you know about item response models. I also assume that you have installed the `dot` program (see page 3 for details).

2 Background

Rosenbaum (1987) considers a set of binary items (here coded 0 or 1) which form a latent scale of a certain class where item characteristic surfaces do not cross. This class includes Guttman, Rasch and the Mokken double monotone scales. He shows that the frequency of occurrence of the item response patterns will form a function decreasing in transposition. In the representation used here a function of the response patterns is decreasing in transposition if rearranging the 0s and 1s so that the 1s are further to the left reduces the value of the function. The sequences of patterns involved can be formed one from another (having the same number of 1s) by transposing a 1 with the 0 immediately to its left.

A convenient way of checking this assumption is by drawing the directed graph in which the nodes are the patterns and an edge exists between patterns connected by the transposition operation. This also provides a useful visual display of the response patterns

3 `latdiag` in action

```
> library(latdiag)
> if(!requireNamespace("ltm")) {
+     warning("You need to install ltm to rebuild the vignette")
+ } else {
+     data(LSAT, package = "ltm")
+ }
> set.seed(12022020)
> res <- draw.latent(LSAT)
> #print(res, rootname = "lsat")
> #plot(res, rootname = "lsat", graphtype = "pdf")
```
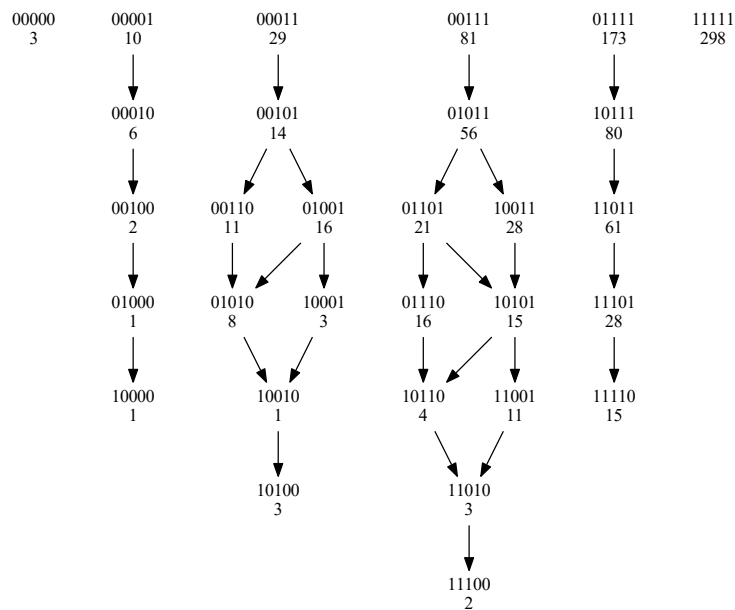

Figure 1: Patterns in the Lsat dataset

An example is shown in Figure 1. This uses the dataset `LSAT` contained in the `ltm` package. To display the patterns properly it is necessary to rearrange the items in order of increasing frequency of 1s and the default print method displays the order which is used for Figure 1.

In Figure 1 only those patterns which occur in the dataset are displayed and this is the default behaviour. The only two possible patterns which did not

in fact occur are 01100 and 11000. We can see that most of the patterns do fit the desired behaviour with some small exceptions. For instance 00101 should be more prevalent than 01001 which is not true (14 versus 16) but the difference from expected is small.

Note that nothing is asserted about the frequencies of patterns with different numbers of 1s for instance 00110 and 10101. Similarly nothing is asserted about patterns not connected in the diagram for instance 00110 and 10001.

We can see that $3 + 10 + 29 + 81 + 173 + 298 = 594$ of the 1000 patterns form a Guttman scale.

3.1 Design decisions

The basic design decision underlying the function is to let the bulk of the detailed graph drawing be undertaken by the specialised graph drawing program `dot` (available from `http://www.graphviz.org/`). Users who wish to undertake fine tuning of their graph can do so by editing the output file of `dot` commands.

3.2 Getting a useful display

By default all patterns which occur are drawn. If there are a large number of items this can lead to an unwieldy display. There are two issues here

1. Some patterns do not lead directly from or to any another pattern (because the nodes which link them have not been printed as they do not occur in the dataset).

2. The diagram becomes very wide

The first of these can be addressed by forcing all patterns to appear even if they occur with frequency zero. The second point can be addressed by selectively printing only those patterns with a certain number of items positive. By doing this successively the whole diagram can be created spread over several pages.

```
> score <- apply(LSAT, 1, mean)
> LSAT$item6 <- sapply(score, function(x) sample(0:1, 1, prob = c(x, 1-x)))
> LSAT$item7 <- sapply(score, function(x) sample(0:1, 1, prob = c(1-x, x)))
> LSAT$item8 <- round(runif(nrow(LSAT)))
> res <- draw.latent(LSAT)
```

```
> #print(res, rootname = "simul")
> #plot(res, rootname = "simul", graphtype = "pdf")
```
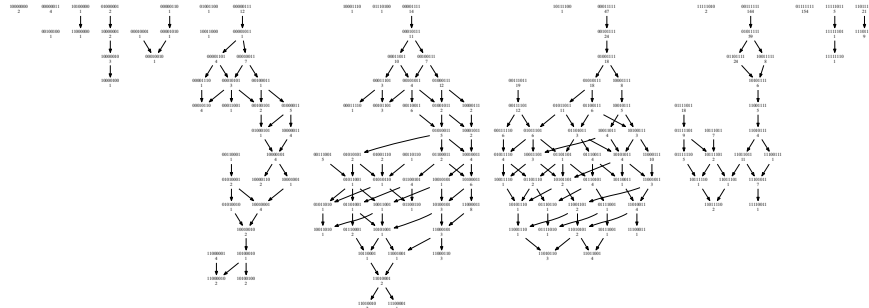
3.3   Fine tuning the plot

Figure 2: Patterns in the Lsat dataset plus three simulated variables

Figure 2 shows what happens if we add three simulated variables to the LSAT data–set. Even with only eight items the plot is difficult to read except at a high zoom factor and then it involves too much panning from side to side.

```
> res <- draw.latent(LSAT, which.npos = 5:6)
> #print(res, rootname = "simul2")
> #plot(res, rootname = "simul2", graphtype = "pdf")
```

We now demonstrate the use of the which.npos parameter to restrict the output to just those patterns with five or six positive. This is shown in Figure 3 where it is clearer to see what is going on. In this we can see a number of cases where we have patterns not decreasing in transposition. For example in the patterns with five positive we have 10100111 occurring 3 times but 1100111 occurs 10 times. Further examples can be found in the patterns with six positives like 11010111 (4 times) followed by 11011011 (11) and also 11100111 (1) followed by 11101011 (7).

3.4   Using dot to draw the final diagram

If you are happy with the defaults you can just call the plot method on the object returned by draw.latent. If you want finer control then assuming you output the dot commands to lsat.gv as shown in
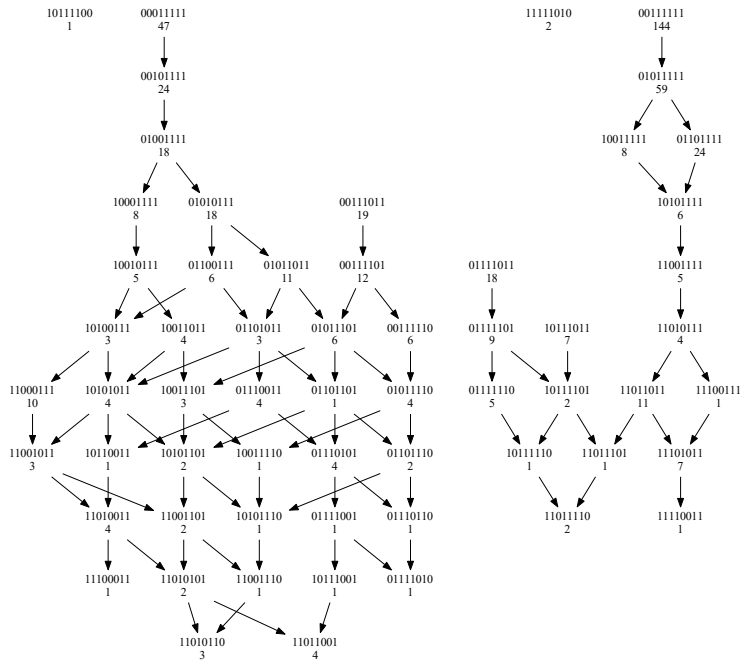
4
```

Figure 3: Patterns in the Lsat dataset

This gives you a file in the desired format in `lsat.pdf`. Many other output formats are possible, see the `dot` documentation for details. Note that I use the extension `.gv` for `dot` commands as this seems the currently preferred file type.

# 4 The theory: Rosenbaum's paper

This section summarises the parts of Rosenbaum (1987) which are relevant for `latdiag`. If you have read Rosenbaum (1987) you do not need to read this section as it adds nothing. I have provided it for the benefit of those who cannot easily obtain Rosenbaum (1987). It does not contain any intellectual contribution from me.

We consider dichotomous items only.

The method considers the largest class of latent variable models for which it can be said that a subset of the items has a single ordering by difficulty that applies to all subgroups of respondents. This includes the Guttman scale, the Rasch model and the Mokken doubly monotone model.

## 4.1 Latent variable models

Let $\mathbf{X}$ be a $J$–dimensional vector of binary responses (1 = correct, 0 = incorrect) to $J$ items, $\mathbf{X} = (X_1, X_2, \ldots, X_j)$, for an individual subject. Let $\mathrm{pr}(\mathbf{X} = \mathbf{x})$ denote the distribution of such response vectors in a specific population of subjects. This is a distribution on the $2^J$ contingency table $X_1 \times X - 2 \times \cdots \times X_J$. A latent variable model for $X$ represents $\mathrm{pr}(\mathbf{X} = \mathbf{x})$ in terms of a latent variable $\mathbf{U}$ which may be a vector.

We usually assume conditional independence of the $J$ item responses given $\mathbf{U}$

$$\mathrm{pr}(\mathbf{X} = \mathbf{x}) = \int \prod_{i=1}^{J} r_i(\mathbf{u})^{x_i} \cdot \{1 - r_i(\mathbf{u})\}^{1-x_i} \mathrm{d}F(\mathbf{u}) \tag{1}$$

where $F(\cdot)$ is the distribution of $\mathbf{U}$ in the given population and $r_j(\mathbf{u}) = \mathrm{pr}(X_j = 1 | \mathbf{U} = \mathbf{u})$ is the item characteristic curve or surface (ICC or ICS) for the $j$th item.

[Section omitted here]

### 4.1.1 ICS which do not cross

One item, $X_j$ is described as uniformly more difficult than $X_i$ if there is an item response representation such that $\forall \mathbf{u}(r_i(\mathbf{u}) \geq r_j(\mathbf{u}))$. In other words the surface for item $i$ lies on or above the surface for item $j$.

We rearrange $\mathbf{X}$ as partitioned into two groups of item $(\mathbf{Y}, \mathbf{Z})$ where $\mathbf{Y}$ contains $K$ items $2 \leq K \leq J$ and $\mathbf{Z}$ contains the remainder. Note that if $K = J$ then $\mathbf{Z}$ is empty. If there is an item response representation such that the items in $\mathbf{Y}$ are ordered by relative difficulty then we shall say that the items in $\mathbf{Y}$ have a latent scale. The items in $\mathbf{Z}$ (if any) are unrestricted. Various types of scale like Guttman, Rasch, and Mokken each impose additional restrictions.

## 5 Functions decreasing in transposition

A function $f(\mathbf{x})$ of a $J$–dimensional vector $\mathbf{x}$ is decreasing in transposition (DT) if rearranging the coordinates of $\mathbf{x}$ so large coordinates are further to the right has the effect of reducing the value of the function. So for three item responses $\mathbf{x} = (x_1, x_2, x_3)$ a function $f(\mathbf{x})$ is DT if

$$
\begin{aligned}
f(100) \geq f(010) &\geq f(001) \quad \text{and} \\
f(110) \geq f(101) &\geq f(011)
\end{aligned}
\tag{2}
$$

Note that nothing is said about whether

$$
\begin{aligned}
f(100) &\geq f(110) \qquad \text{or} \\
f(100) &< f(110)
\end{aligned}
\tag{3}
$$

as they are not connected by a single permutation. Similarly nothing is said about $f(000)$ or $f(111)$.

For four item responses $\mathbf{x} = (x_1, x_2, x_3, x_4)$

$$
\begin{aligned}
f(1100) \geq f(1010) &\geq f(1001) \geq f(0101) &\geq f(0011) \\
f(1100) \geq f(1010) &\geq f(0110) \geq f(0101) &\geq f(0011)
\end{aligned}
\tag{4}
$$

but we may have either $f(0110) < f(1001)$ or $f(0110) > f(1001)$ because neither 0110 nor 1001 can be obtained from the other by moving a 1 to the right and a 0 to the left.

[Section omitted here]

### 5.0.1 Some examples of functions DT

$$f_E(\mathbf{x}) = \sum_{i=1}^{I} x_i \tag{5}$$

$f_E(\cdot)$ is the number of the correct responses among the first $I$ of the $J$ responses, $I < J$. If two items with $i, j \leq I$ are transposed then it does not alter $f_E$ nor if $i, j > I$ but if $i \leq I$ and $j > I$ and $x_i = 1$ and $x_j = 0$ then it does as it reduces $f(\mathbf{x})$ by 1. Note that

$$f_D(\mathbf{x}) = \sum_{i=I+1}^{J} x_i \tag{6}$$

is not DIT but $J - f_D(\mathbf{x}) = f_E(\mathbf{x})$ is.

If we have weights, $w_1 \geq w_2 \geq \cdots \geq w_J$ then

$$f_W(\mathbf{x}) = \sum_{i=1}^{J} w_i \cdot x_i \tag{7}$$

Of course $f_E(\mathbf{x})$ may be written in this form with $w_i = 1$ for $i \leq I$ and $w_i = 0$ elsewhere. If the items are ranked by difficulty (most difficult with rank 1, least with rank $J$), ties given average rank and those ranks used as the weights $w$ then $f_W(\mathbf{x})$ is the sum of the ranks of the items answered correctly and resembles the Wilcoxon rank–sum statistic.

Finally consider the indicator function $[f(\mathbf{x}) \geq k]$ which equals 1 if $f(\mathbf{x}) \geq k$ and zero otherwise. If $f(\mathbf{x})$ is DT then so is $[f(\mathbf{x}) \geq k]$.

## 5.1 Properties of observable distributions when Y las a latent scale

If $\mathbf{Y}$ has a latent scale then the conditional distribution of $\mathbf{Y}$ given any function of $\mathbf{Z}$ is decreasing in transposition, ie

$$\mathrm{pr}\{\mathbf{Y} = \mathbf{y} | \mathbf{h}(\mathbf{Z})\} \geq \mathrm{pr}\{\mathbf{Y} = \mathbf{y}^* | \mathbf{h}(\mathbf{Z})\} \tag{8}$$

whenever $\mathbf{y}^*$ can be obtained from $\mathbf{y}$ by interchanging two coordinates, $y_i$ and $y_j$, with $y_i = 1$, $y_j = 0$ and $i < j$. In the important special case where $\mathbf{Z}$ is empty this implies that $\mathrm{pr}\mathbf{X} = \mathbf{x}$ is DT. When $\mathbf{Z}$ contains many items a suitable choice for $\mathbf{h}(\mathbf{Z})$ is the total score on items in $\mathbf{Z}$.

# References

P R Rosenbaum. Probability inequalities for latent scales. *British Journal of Mathematical and Statistical Psychology*, 40:157–168, 1987.