Identify connected subsets of the data

Gregor Gorjanc
gregor.gorjanc@bfro.uni-lj.si

March 4, 2007

1 Introduction

R package *connectedness* provides functions to identify (dis)connected subsets in the data (Searle, 1987). Current implementation finds disconnected sets in a two-way classification without interaction as proposed by Fernando et al. (1983).

2 Demo

2.1 Data

Package ships with an example dataset `connect`, which contains two variables: `group` and `season`. This dataset is really small, but big enough to show the idea.

```
> library(connectedness)
> data(connect)
> connect
```

```
   group season
1      G     NA
2      A      1
3      A      3
4      B      1
5      B     NA
6      C      2
7      D      2
8      E      2
9      E      4
10     F      3
11     G      4
12     H     NA
13     G     NA
14  <NA>      1
```

```
> table(connect$group, connect$season)

    1 2 3 4
  A 1 0 1 0
  B 1 0 0 0
  C 0 1 0 0
  D 0 1 0 0
  E 0 1 0 1
  F 0 0 1 0
  G 0 0 0 1
  H 0 0 0 0
```

As can be seen, some values are missing and cross-tabulation clearly shows that this design is unbalanced. Instead of looking hard into the cross-tabulation table we can use `connectedness()` function, which will tell us if there are any disconnected subsets in this data. A subset is defined as a part of the data, where partitioning is based on factor levels. Things are simple in one dimension, but complicate when more factors are involved. Data is said to be connected, if we can connect cells in a cross-table in such a way that all “filled” cells are connected (Searle, 1987). Example will demonstrate this in turn.

2.2 `connectedness()`

Function `connectedness()` accepts variables and returns object of class *connectedness*. Its print method shows number of disconnected subsets and “summary” by each subset: number of records and levels or each factor. Example dataset has 3 disconnected subsets. Third subset is a bit special as it has only one level of `group` factor, but none for `season` factor - all non-factor variables are converted to factors. Such subsets can be removed by use of `drop=TRUE` in call to `connectedness` - check its help page for the details.

```
> tmp <- connectedness(x = connect$group, y = connect$season)
> tmp

Connectedness between: connect$group connect$season
Number of disconnected subsets: 3
Subsets:
  Subset Freq   Percent Levels1 Levels2
1      2    7 50.000000 C D E G     2 4
2      1    6 42.857143   A B F     1 3
3      3    1  7.142857       H
```

2.3 `plot()`

Since picture tells more than hundred words, there is also a plot method. It accepts object of class *connectedness*. It has various arguments to control the plot. Check its help page for the details. We will use arguments *lines=TRUE*, *linesSubset=1* and *linesArg=list(col=”black”, lwd=2)* to add a line to the plot for subset 1 to show the intuitive meaning of connectedness. For better display we increase the width of the line and set its colour to black. Third subset is not shown on the picture, since it does not have any data on second factor. Read help page for the details.

```
> plot(tmp, lines = TRUE, linesSubset = 1, linesArg = list(col = "black",
+     lwd = 2))
```

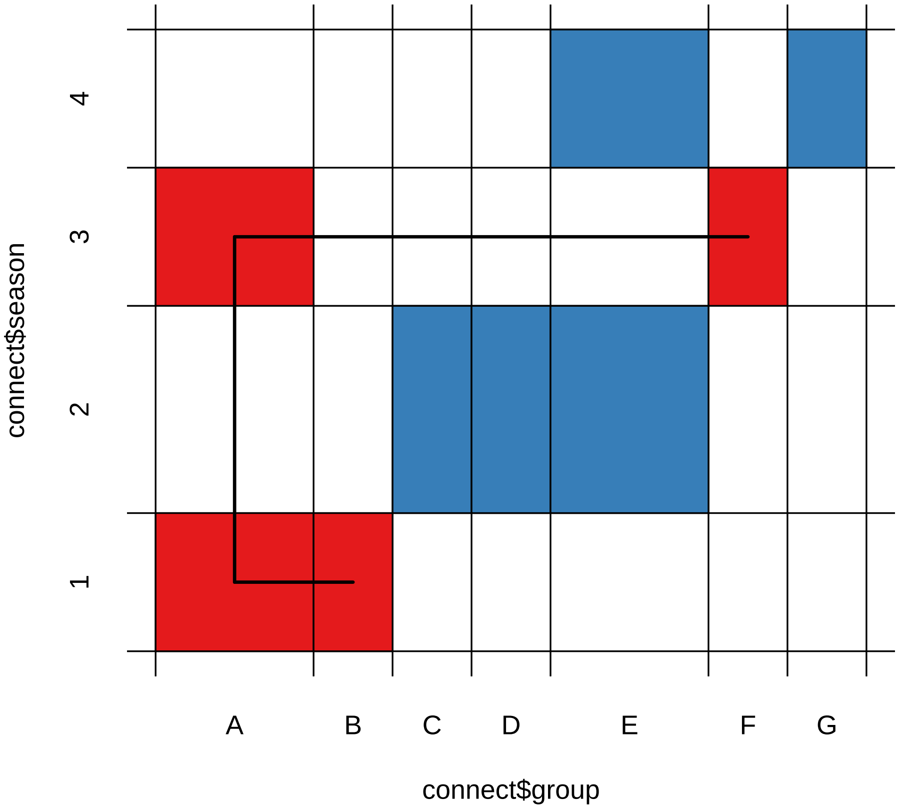

Figure 1: Graphical display of subsets in the data based on group and season factors; grid shows distribution of data, colours differentiate subsets and line represents the intuitive meaning of connectedness

2.4 Other utilities

Now we know that there are three subsets in our data. If we want to get levels for factors by each subset, we can use `levelsBySubset` function.

```
> levelsBySubset(x = tmp)
```

```
$`2`
$`2`$Levels1
[1] "C" "D" "E" "G"
```

```
$`2`$Levels2
[1] "2" "4"


$`1`
$`1`$Levels1
[1] "A" "B" "F"

$`1`$Levels2
[1] "1" "3"


$`3`
$`3`$Levels1
[1] "H"

$`3`$Levels2
character(0)
```

If we want to use mentioned factors in our statistical model, we can use information from connectedness analysis. Namely, disconnectedness causes problems in estimability. `subset()` can be used to subset the data and we can then perform separate analyses per subset.

```
> subset(x = tmp, data = connect, subset = 1)

   group season
2      A      1
3      A      3
4      B      1
5      B     NA
10     F      3
14  <NA>      1
```

3 Future plans

Current functionality and implementation is light and neat due to simplicity of the used algorithm by Fernando et al. (1983). However, it is limited to two factors only. Further versions will try to cope with more general approaches as well as to quantify the degree of connectedness between levels. Please do not hesitate to contact me in case of suggestions, contributions, bugs or enthusiasm to take over the development of the package.

4 *R* Session information

```
> toLatex(sessionInfo())
```

- R version 2.4.1 (2006-12-18), `i486-pc-linux-gnu`
- Locale: `LC_CTYPE=en_GB.UTF-8;LC_NUMERIC=C;LC_TIME=en_GB.UTF-8;LC_COLLATE=en_GB.UTF-8;LC_MONETARY=en_GB.UTF-8;LC_MESSAGES=en_GB.UTF-8;LC_PAPER=en_GB.UTF-8;LC_NAME=C;LC_ADDRESS=C;LC`
  `8;LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: connectedness 0.2.2

References

Fernando, R. L., Gianola, D., and Grossman, M. (1983). Identify all connected subsets in a two-way classification without interaction. *J. Dairy Sci.*, 66:1399–1402.

Searle, S. R. (1987). *Linear models for unbalanced models.* John Wiley & Sons.