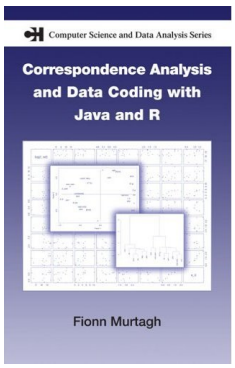
Review of Fionn Murtagh's book: Correspondence Analysis and Data Coding with Java and R

by Susan Holmes



Correspondence analysis is a method for analysing contingency tables, in particular two-way tables, by decomposing the Chi-square distances between the rows in much the same way that principal component analysis (PCA) decomposes the variance of a multivariate continuous matrix. For the reader familiar with multidimensional scaling (Mardia et al., 1979; Ripley, 1996), this can also be seen as a way of representing Chi-square distances between the rows or the columns of the contingency table. This method has been rediscovered regularly and interested readers can read the history of the method in Benzécri (1982) and de Leeuw (1983).

For those familiar with the matrix formulation of PCA as the singular value decomposition of $X_{n\times p}$ as the product of orthogonal matrices U and V with a diagonal matrix of decreasing singular values S, $X = USV'$, this provides the best rank k approximation to X by just taking the first k columns of U. To get correspondence analysis, replace X by the contingency table divided by its total (call this F, the frequency table), call the row sums of F $\mathbf{r}$ and the column sums $\mathbf{c}$. Let the diagonal matrices defined by these vectors be $\mathbf{D}_r$ and $\mathbf{D}_c$. Write the singular value decomposition of

$$\mathbf{D}_r{}^{-\frac{1}{2}} F \mathbf{D}_c{}^{-\frac{1}{2}} = USV'$$

with $V'V = I, U'U = I$; then $\mathbf{D}_r{}^{-1} F \mathbf{D}_c{}^{-1}$ can be written ASB', with $A = \mathbf{D}_r{}^{-\frac{1}{2}} U$ and $B = \mathbf{D}_c{}^{-\frac{1}{2}} V$. One can verify that $A'D_rA = I$ and $B'D_cB = I$; see Holmes (2006) for more details. This decomposition can be used to provide a graphical representation of both columns and rows that is meaningful even when both are plotted on the same graph. This biplot representation is a useful feature of the method.

What the book contains

The book starts by setting the stage for the book's particular perspective with a foreword by Jean-Paul Benzécri, a leader in the world of geometric approaches to multivariate analyses. Chapter 2 covers the advent of France's particular school of what was called 'Analyse des Données' to distinguish it from classical parametric statistics based on probability theory. The insistence of Murtagh's mentor, Jean-Paul Benzécri on the data over the model is a precursor to modern day machine learning and a European parallel to John Tukey's Exploratory Data Analysis.

After the introduction, chapter 3 gives a complete matrix-based explanation of correspondence analysis followed by chapter 4 on the importance of choosing the correct encoding of data, and chapter 5 presents examples. Chapter 6 is dedicated to the specific study of textual analyses.

Fionn Murtagh aims to explain how to effectively analyse multivariate categorical data. He presents both the different ways of encoding the data and ways of analysing them following the French approach to correspondence analysis. Many parts of the book are also of historical interest: the introduction by Benzecri and the fact that the programs were coded from his original Pascal programs for instance. His text contains many examples especially from the field of textual analysis, closely following his mentor.

The Rebirth of Correspondence Analysis

The book is somewhat old fashioned and doesn't present bridges to today's literature. Correspondence analysis is regularly rediscovered, and today's literature on the decomposition of the structure of the web is no exception. An interesting connection between Kleinberg's hubs and authorities (Kleinberg, 1999) and correspondence analysis is pointed out by Fouss et al. (2004). This should motivate modern readers to understand something that lies at the heart of successes such as Google's page-rank system. In fact, a good example of what profiles are can be provided by looking at Google's Trends feature, for which the towns are given as profiles divided by the number of searches for that city. Try googling yourself [1].

[1] http://www.google.com/trends?q=statistical+software\&ctab=0\&geo=all\&date=2005

Missing Topics

It's a shame that the author didn't take the opportunity to explain in more detail to the English speaking audience the French community's special tricks for interpreting correspondence analysis. Although there are special programs for supplementary variables and individuals in the book (from page 59 on), and they are mentioned on pages 41 and 43, no example shows clearly how they are used; this would have been an important contribution to the existing literature.

Important Caveats

The uninitiated user should be cautioned as to how important it is to compare eigenvalues at the outset of the analysis. Scree plots of the eigenvalues have to be made before choosing the number of axes, and the big mistakes that occur in CA happen when two close eigenvalues are split. There may be instances when not separating three axes with three similar eigenvalues would encourage use of tools such as `xgobi` and `ggobi` for dynamic multidimensional graphics.

The book would also be improved by more background about special aspects of correspondence analysis, such as Guttman's horseshoe effect and seriation (Charnomordic and Holmes, 2001), for which correspondence analysis is especially useful.

The present reviewer's frustration with this book comes mostly from the fact that there are many good 'First Books on Correspondence Analaysis' available in English (Greenacre, 1984; Lebart et al., 1984). There are some first-rate R packages for performing correspondence analysis that we list at the end of this review. However there is no good followup text that carefully explains useful techniques developed in the literature such as internal correspondence analysis (Cazes et al., 1988), asymmetric correspondence analysis, conjoint analysis of several contingency tables through ACT (Lavit et al., 1994) or through cocorrespondence analysis (ter Braak, 1986).

Software implementation

As regards the software implementation provided, it does not seem a good use of paper to present the source code in the main text, but providing a complete R package on CRAN would definitely be a worthwhile investment. The R code as it exists is not modernised and doesn't use classes or object oritentation.

The Java code was not useful to the present reviewer as on our modern machine (Mac running Mac OS X 10.4) the instructions provided on the website do not allow one to execute the JAVA downloaded.

Packages containing other R correspondence analysis programs available at CRAN

`ade4` : The most complete suite of multivariate functions 'à la française', this package includes internal correspondence analysis, asymmetric correspondence analysis, detrended correspondence analysis and many visualization tools. Its philosophy and class structures are based on the triplets defined as duality diagrams in Escoufier (1977) and discussed more recently (Holmes, 2006). A description of the simpler methods (one table methods) is available in English in Chessel et al. (2004).

`amap` The function `afc` performs the classical correspondence analysis.

`CoCorresp` Package for doing correspondence analysis on several tables and co-inertia analyses especially used by ecologists.

`ca` **and** `homals` from Jan de Leeuw's Psychometrics with R bundle (`http://www.cuddyvalley.org/psychoR/`).

`vegan` Developed by plant ecologists, this package allows one to perform constrained as well as vanilla correspondence analysis in the function `cca`; it also provides much needed detrending for gradients in the `decorana` function, which implements Hill and Gauch's detrending technique (Hill and Gauch, 1980).

`VR` Venables and Ripley bundle containing MASS. Within MASS is the bare-bones function `corresp` which allows one to produce a correspondence analysis map and use the `biplot` function to plot it.

Bibliography

J.-P. Benzécri. *Histoire et préhistoire de l'analyse des données*. Dunod, 1982.

P. Cazes, D. Chessel, and S. Dolédec. L'analyse des correspondances internes d'un tableau partitionné: son usage en hydrobiologie. *Revue de Statistique Appliquée*, 36:39–54, 1988.

B. Charnomordic and S. Holmes. Correspondence analysis for microarrays. *Statistical Graphics and Computing Newlsetter*, 12, 2001.

D. Chessel, A. B. Dufour, and J. Thioulouse. The ade4 package — I: One-table methods. *R News*, 4(1):5–10, 2004. URL `http://CRAN.R-project.org/doc/Rnews/`.

J. de Leeuw. On the prehistory of correspondence analysis. *Statistica Neerelandica*, 37:161–164, 1983.

Y. Escoufier. Operators related to a data matrix. In J. R. Barra, editor, *Recent Developments in Statistics.*, pages 125–131. North Holland, 1977.

F. Fouss, J.-M. Renders, and M. Saerens. Some relationships between Kleinberg's hubs and authorities, correspondence analysis, and the Salsa algorithm. In *JADT 2004, International Conference on the Statistical Analysis of Textual Data*, pages 445–455, Louvain-la-Neuve, 2004.

M. J. Greenacre. *Theory and Applications of Correspondence Analysis*. Academic Press, London., 1984.

M. Hill and H. Gauch. Detrended correspondence analysis, an improved ordination technique. *Vegetatio*, 42:47–58, 1980.

S. Holmes. Multivariate analysis: The French way. In *Festschrift for David Freedman*, Beachwood, OH, 2006. (edited by D. Nolan and T. Speed) IMS.

J. M. Kleinberg. Hubs, authorities, and communities. *ACM Computing Surveys*, 31(4es):5, 1999. ISSN 0360-0300.

C. Lavit, Y. Escoufier, R. Sabatier, and Traissac. The ACT (STATIS method). *Computational Statistics & Data Analysis*, 18:97–119, 1994.

L. Lebart, A. Morineau, and K. M. Warwick. *Multivariate Descriptive Statistical Analysis*. Wiley, 1984.

K. Mardia, J. Kent, and J. Bibby. *Multiariate Analysis*. Academic Press, NY., 1979.

B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK, 1996.

C. J. F. ter Braak. Canonical correspondence analysis: a new eigenvector method for multivariate direct gradient analysis. *Ecology*, 67:1167–1179, 1986.

Susan Holmes
Statistics Department
Stanford, CA 94305
susan@stat.stanford.edu

R Help Desk

Accessing the Sources

by Uwe Ligges

Introduction

One of the major advantages of open source software such as R is implied by its name: the sources are open (accessible) for everybody.

There are many reasons to look at the sources. One example is that a user might want to know exactly what the software does, but the documentation is not sufficiently explicit about the underlying algorithm. As another example, a user might want to change some code in order to implement a modification of a given (already implemented) method or in order to fix a bug in a contributed package or even in R itself.

How to access different kinds of sources (in order to read or to change them), both in R itself and in packages, is described in the following sections.

It is always a good idea to look into appropriate manuals for your current R version, if working on the sources is required. Almost all manuals contain relevant information for this topic: 'An Introduction to R', 'Writing R Extensions', 'R Installation and Administration', and 'The R Language Definition' (Venables et al., 2006; R Development Core Team, 2006a,b,c).

R Code Sources

In most cases, it is sufficient to read some R code of the function in question and look at how other functions are called or how the data are manipulated within a function. The fastest and easiest way to do so for simple functions is to type the function's name into R and let R print the object. For example, here is how to view the source code for the function `matrix()`:

```
> matrix
function (data = NA, nrow = 1, ncol = 1,
          byrow = FALSE, dimnames = NULL)
{
    data <- as.vector(data)
    if (missing(nrow))
        nrow <- ceiling(length(data)/ncol)
    else if (missing(ncol))
        ncol <- ceiling(length(data)/nrow)
    x <- .Internal(matrix(data, nrow, ncol,
                          byrow))
    dimnames(x) <- dimnames
    x
}
<environment: namespace:base>
```

Unfortunately, comments in the code may have been removed from the printed output, because they were already removed in the loaded or installed package in order to save memory. This is *in principle* controlled by the arguments `keep.source` (for R) and