Interactive biplot construction

Frederic Udina <udina@upf.es>*
Departament d'Economia i Empresa
Universitat Pompeu Fabra

June 4, 2004

Abstract

We analyze and discuss how a generic software to produce biplot graphs should be designed. We describe a data structure appropriate to include the biplot description and we specify the algorithm(s) to be used for several biplot types.

We discuss the options the software should offer to the user in two different environments. In a highly interactive environment the user should be able to specify many graphical options and also to change them using the usual interactive tools. The resulting graph needs to be available in several formats, including high quality format for printing. In a web-based environment, the user submits a data file or listing together with some options specified either in a file or using a form. Then the graphic is sent back to the user in one of several possible formats according to the specifications.

We review some of the already available software and we present an implementation based in Xlisp-Stat. It can be run under Unix or Windows, and it is also part of a service that provides biplot graphs through the web.

1 Introduction

Biplots (Gabriel (1971), Gower and Hand (1996)), can be seen as the multivariate analogue of scatter-plots: they give a graphical represen-

*Michael Greenacre contributed with many initial ideas and fruitful discussion. Support of Spanish Government Grant BMF-2003-03324 is acknowledged.

tation of a multivariate sample and they superimpose on the display a representation of the variables on which the sample is measured. Biplots are useful both in the data exploration and data description phases of each data analysis. This implies that biplot production has to be interactive and that good quality output is also needed. As Gower and Hand (1996) say, page 27,

> The notion of inspecting, and perhaps discarding, biplot axes, implies the desirability of developing interactive software with appropriate facilities.

In this paper we describe how statistical software to produce biplots should be designed and we introduce XLS-Biplot, an XLisp-Stat (Tierney (1990)) package that includes most of the desired features. A full description of XLS-Biplot may be found in the user manual included in the distribution. It is also available through the WWW, see the Appendix.

In Section 2 we describe the computations involved in biplot construction in several statistical models and we discuss some of the requirements for a biplot software.In Section 3 we review some other available software and we explain why Xlisp-Stat was the right choice for a first implementation of the program. In Section 4 we introduce XLS-Biplot. In the Appendix we include some technical information on how to obtain and install the software.

We do not include a full discussion of the different statistical methods than produce biplots. As the biplot has different interpretations and uses in these different cases, a specialized work such as Gower and Hand (1996) should be consulted.

2 Computations

The general layout of the computations is shown in Figure 1

2.1 Data matrices

The data matrix $\mathbf{X}$ can be either

- cases $\times$ variables. We will always assume cases are rows, and variables are columns.
- categories $\times$ categories, a cross tabulation or other table of counts;
- cases $\times$ categories, compositional data.

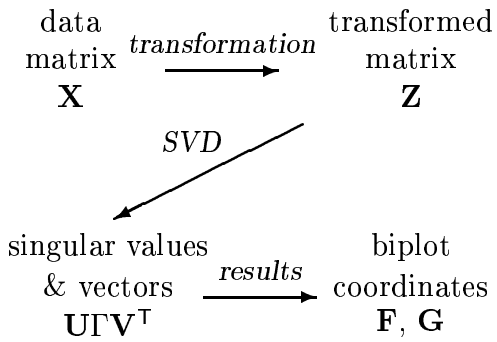

Figure 1: Layout of computations involved in biplot construction

At the moment we do not consider other settings that may use a biplot for data visualization, such as multiple, joint or canonical correspondence analysis or redundancy analysis.

2.2 Transformations

Allowed transformations (it is important to apply these transformations in the sequence specified below):

- functional transformations such as logarithmic, sine, power.
- matrix transformations such as observed/expected frequency ratio (Pearson's contingency ratio), odds ratio, logratio, ...;
- application of row and/or column weights;
- (weighted) centering of rows or columns;
- (weighted) double-centering;
- (weighted) normalization of rows or columns.

2.3 SVD

We perform (weighted) Singular Value Decomposition (SVD) via the eigendecomposition of a cross-product matrix whose order depends on the number of columns of $\mathbf{Z}$ (i.e., $\mathbf{Z}^\top\mathbf{Z}$, since the number of columns is usually less than the number of rows – if this is not so, then we will

merely be calculating a larger than necessary matrix and using more computational time for the solution.

The most general situation we shall need is when there are weights $\mathbf{r}$ for the rows and weights $\mathbf{c}$ for the columns, with associated diagonal matrices $\mathbf{D}_r$ and $\mathbf{D}_c$. Defaults are $\mathbf{D}_r = (1/n)\mathbf{I}$ (each of the n rows is weighted equally by $1/n$) and $\mathbf{D}_c = \mathbf{I}$ (Notice that to be consistent, we should weight the p columns by $1/p$, which would mean – in the case of PCA – that we decompose the *average variance*, not the *total variance*. This would make more sense but is never done in practice).

The weighted SVD is obtained by decomposing the matrix

$$\mathbf{D}_r^{1/2}\mathbf{Z}\mathbf{D}_c^{1/2} = \mathbf{U}\Gamma\mathbf{V}^{\mathsf{T}} \tag{1}$$

where Γ is the diagonal matrix of eigenvalues. Alternatively, if proceeding via the weighted eigendecomposition,

$$\mathbf{D}_c^{1/2}\mathbf{Z}^{\mathsf{T}}\mathbf{D}_r\mathbf{Z}\mathbf{D}_c^{1/2} = \mathbf{V}\Gamma^2\mathbf{V}^{\mathsf{T}}$$

where Γ^2 is the diagonal matrix of eigenvalues, followed by the transformation to the left vectors:

$$\mathbf{U} = \mathbf{D}_r^{1/2}\mathbf{Z}\mathbf{D}_c^{1/2}\mathbf{V}\Gamma^{-1}$$

2.4 Computation of coordinates

There are three types of coordinates, and four standard biplot/map options.

Standard coordinates:

$$\text{Rows}: \quad \mathbf{F}_s = \mathbf{D}_r^{-1/2}\mathbf{U} \qquad \text{Columns}: \quad \mathbf{G}_s = \mathbf{D}_c^{-1/2}\mathbf{V}$$

Principal coordinates:

$$\text{Rows}: \quad \mathbf{F}_p = \mathbf{F}_s\Gamma \qquad \text{Columns}: \quad \mathbf{G}_p = \mathbf{G}_s\Gamma$$

"Canonical" coordinates:

$$\text{Rows}: \quad \mathbf{F}_c = \mathbf{F}_s\Gamma^{1/2} \qquad \text{Columns}: \quad \mathbf{G}_c = \mathbf{G}_s\Gamma^{1/2}$$

Asymmetric map (form biplot) of the rows: plot $\mathbf{F}_p$ and $\mathbf{G}_s$.
Asymmetric map (covariance biplot) of the columns: plot $\mathbf{F}_s$ and $\mathbf{G}_p$.
Symmetric map (not a biplot): plot $\mathbf{F}_p$ and $\mathbf{G}_p$.
Symmetric or canonical biplot: plot $\mathbf{F}_c$ and $\mathbf{G}_c$.

2.5 "Canned" analysis

These are predefined sequences for certain types of data.

- PCA without normalization: centering of columns → SVD, etc...
- PCA with normalization: centering of columns → normalization of columns → SVD, etc...
- simple CA: transform to Pearson contingency ratios → apply row and column masses → (weighted) double-centring → (weighted) SVD, etc...
- logratio analysis (Aitchison): transform to logarithms → double-centring → SVD, etc...
- ratio maps (Greenacre, or spectral mapping, Lewi): transform to logarithms → apply row and column masses → (weighted) double-centring → (weighted) SVD, etc...

2.6 Building the biplot

Once the coordinates of the cases (rows) and the variables (columns) are computed according to the previous sections, some important questions should be considered for the biplot to be correct and fully interpretable. The main one is that the scales used on the axes of the graph should be identical. Otherwise, distances and angles (including orthogonal projections) are not preserved and the interpretation of the graph will be incorrect.

The two main axes in the biplot are determined by the orthogonal directions of maximum variance or variability. The choice of the positive direction in those axes is quite arbitrary and it is very convenient to allow the user to choose it. That is, the software should give the user some way to reverse the axis directions once the initial solution has been obtained.

It is important to know, given a biplot, the *quality of representation* the biplot is giving. In most cases this is achieved by measuring the percentage of the total variance or variability of the original data that is represented in the graph, and it can be computed as the percentage of the eigenvalues produced in (1) corresponding to each one of the two axes present in the bi-dimensional biplot, relative to the total sum of the eigenvalues. So the program should give the amount of variability captured by each axis as well as the sum of two axes being biplotted. More precisely, these quantities are as follows, assuming

that $\lambda_1, \lambda_2, \ldots, \lambda_p$ are the eigenvalues obtained in (1) and that we want the quality for the biplot of dimensions 1 and 2,

$$q_i = \frac{\lambda_i}{\sum_j \lambda_j} \quad (2)$$

$$Q = q_1 + q_2 \quad (3)$$

Cases (rows) are represented as points in the biplot or simply by their label. It is convenient to give the user a choice of symbols sizes and colors to represent points, and this should be possible to automate through specifications in the input data file.

Variables (columns) can be represented in several ways: point symbols, arrows or full length lines. What is the more appropriate depends on the statistical procedure being used. The length of the arrows (or the final coordinates of the points) representing variables comes straight from the computations described in Section 2.4 according to the type of biplot/coordinates selected by the user. In many cases, the arrow lengths is relevant only relative to one another and it may be convenient to modify all the lengths proportionally, an option the software should allow. Changing the length of a single arrow should not be allowed.

In some cases (see for example Gower and Hand (1996), p 26) drawing axes to represent variables will be preferred. In this case, it is also desirable to draw scaled axes showing the values of the variable being represented, a process called calibration. The main problem with this option is that the plot can be overcrowded. One convenient solution is to draw only small tick marks on the variable axes and show the value only when the mouse is over the tick mark.

As a way to build richer biplots, a good tool could be the use of *layers*. Layers are groups of cases and/or variables that share some properties and that can be included/discarded from the biplot at once. For example, through the use of layers it should be easy to assign color, a special symbol or a drawing style to points and arrows on a given layer, or to draw a line connecting all points in a layer. It should be possible to define a layer by enumeration but also by criteria based on variable values or the quality of representation of the elements in the biplot. Layers should be easily shown/hidden.

3 Implementation

3.1 Software Review

We do not want to make here a review of all the statistical packages available, but we want to mention a couple of papers that deal with biplots.

Lipkovich and Smith (2002) presents a series of Microsoft Excel macros to draw biplots from data in a spreadsheet. The strongest point of this package is obviously that it runs in a widely used platform. It is also very complete, as it does biplots for principal components analysis, correspondence analysis, canonical discriminant analysis, metric multidimensional scaling, redundancy analysis, canonical correlation analysis and canonical correspondence analysis. The weakest point of these macros is that they do not force the scales in the two axis to be equal and thus the graphs it produces are not interpretable unless the user does the job of getting the right aspect ratio. Nothing in the software or the documentation does mention this point. Another problem with the macros is that the interactive modification of the elements of the graph is quite difficult, and the production of good quality graphical output is not easy.

Bond and Michailides (1997) includes an extensive study on Homogeneity Analysis (that is similar to Correspondence Analysis) and uses some biplots in the examples, but the goal of the paper is not biplot building and therefore they not provide interactive tools to modify the graph or good quality graphical output.

Most of the more popular statistical packages have some biplot building capability, many of them in the form of contributed macros or extensions. Many of them suffer too the same problem mentioned before: the graphical scales in the biplot axis are not forced to be equal. We know no package that may produce a correct biplot that can be modified interactively to fit the user's needs and that can be converted to a good quality output format. SPSS, for example, can build biplots for PC, CA and also for MDS, but only on some of these cases the biplot is correct, and in no case is easy to modify the biplot. The PC biplot, for example, does not represent the variables as axis or arrows, making it difficult to interpret the graph. For the SAS system, as shown by Friendly (2000), there are good SAS/IML procedures available to produce biplots, but interaction with them is not allowed.

3.2 Why Xlisp-Stat?

The environment to develop graphical software like the one we are describing need to have a number of features.

- It should be object oriented to allow easy programming of a consistent user interface and computational scheme.
- It should allow for fast prototyping and easy program maintenance.
- It should include easy to program interaction tools (mouse control, dialog windows, menus, etc.)
- It should run in a variety of platforms.
- Free software based should be preferred to not restrict potential users.

Xlisp-Stat has all these features and we know of no other statistical package with a so versatile graphical interactive tools that may run on all the major operating systems. This is the reason of the choice. But it is also clear that Xlisp-Stat is a quite old environment and it has no evolved in the last years to be in touch with some of the needed features. As regarding XLS-Biplot, the main limitations of the GUI support in Xlisp-Stat are: there is no font control in the graphical windows, color support is quite limited, and access to the system's clipboard is very poor.

4 XLS-Biplot

In this section we give an overview of XLS-Biplot, the Xlisp-Stat implementation of the software described in the previous sections. Detailed information at the user level may be found in the User's Manual (see the Appendix).

4.1 Some examples

To show the features available in XLS-Biplot we include some examples in the figures below. We deliberately mix in those figures several options that are not desirable in the same biplot.

Figure 2 shows a Principal Components Biplot for a data set that consists of the percentage in each sector of the economy of several European countries. To build the biplot as it is shown, the horizontal

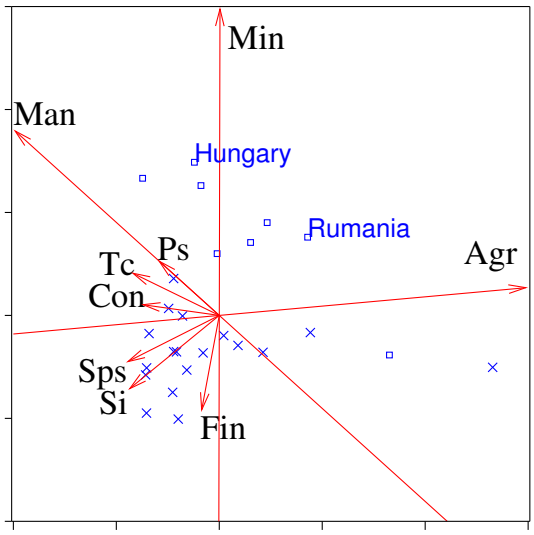

Figure 2: Some examples of the features included in XLS-Biplot

axis has been reversed (to show better the main direction associated with "Agr" (agriculture)). Then, some of the variables have been drawn "as axes" to make it easier to project points orthogonally onto them. The points belonging to formerly communist countries have been given a square as symbol to distinguish them from the rest that have an "X" symbol. Finally, some countries have its label displayed and other do not.

Figure 3 shows a correspondence analysis biplot representing the data set discussed in Greenacre (1993), page 21. The data concern 312 people in a readership survey who have been classified firstly into one of five education groups (E1 to E5) and secondly into one of three levels of readership of a certain newspaper (C1 to C3)

Figure 4 shows a XLS-Biplot window. The window has three main parts: the graph, the Tool menu at the left and the message area at the bottom. As the mouse points to different parts of the window the appropriate message is shown in the message area.

4.2 Overview of features

In this section we summarize the features already implemented in the current version of XLS-Biplot among those described in section 2.

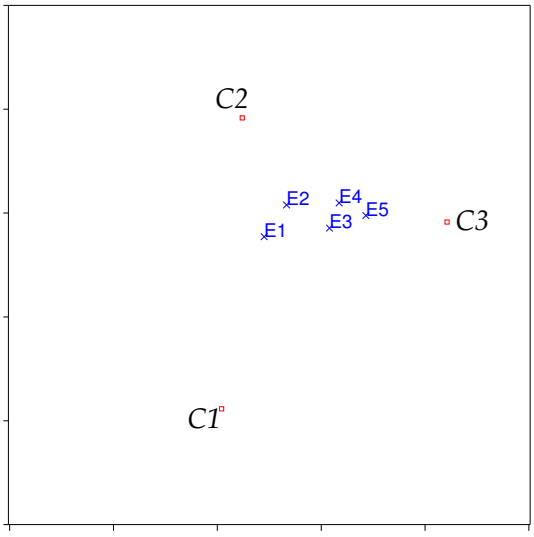

Figure 3: Another example of the features included in XLS-Biplot

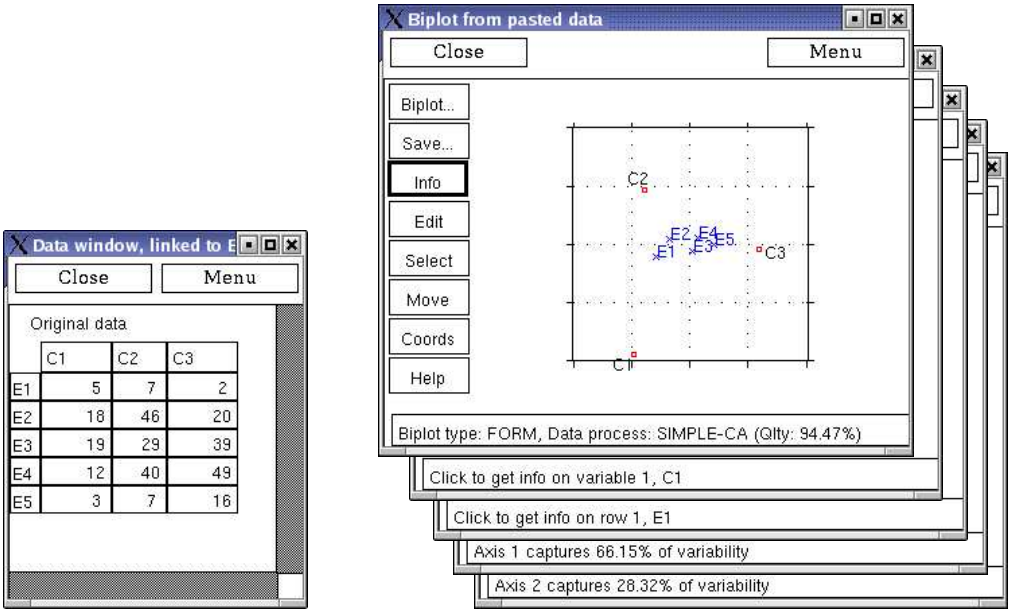

Figure 4: A biplot as it is shown in the XLS-Biplot window (right) and a window showing the biplot data (left). In the lower part of the biplot window, several messages may appear, depending on what element the mouse is pointing.

4.2.1 Data input

XLS-Biplot can make a biplot starting either from a simple data array, a data array with labels in the first column and/or in the first row, or from a special format that can convey all the needed information. The format is fully specified in the user manual and includes provision for specifying variable types, label and weights; case labels and weights, kind of data processing to apply and type of biplot to build. In the current version, no provision is made to specify layers or other information on covariates, extra cases or lines joining cases.

4.2.2 Data processing

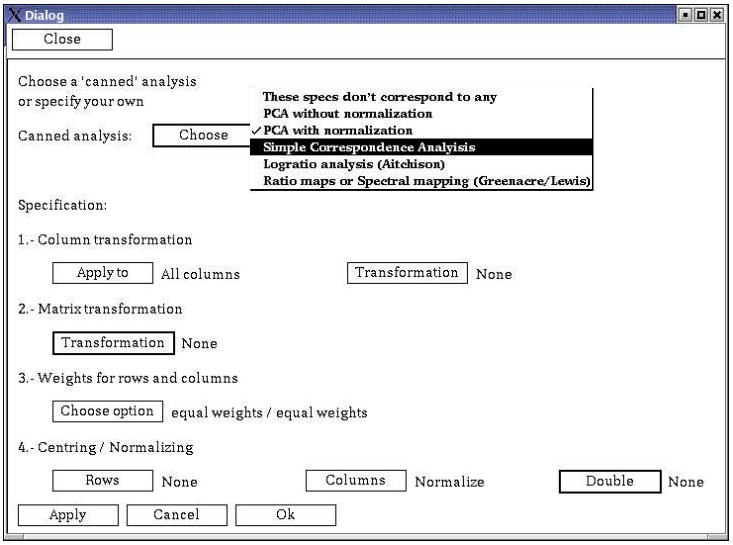

Figure 5: Dialog window used in XLS-Biplot to specify data processing.

The process to apply to the data before singular value decomposition is specified in XLS-Biplot using the dialog window shown in Figure 5. The user may use the menu in the upper part to specify one of the five *canned* analysis described in section 2.2. Alternatively, the user may use the lower part of the dialog window to specify how to do each one of the four phases of the transformation.

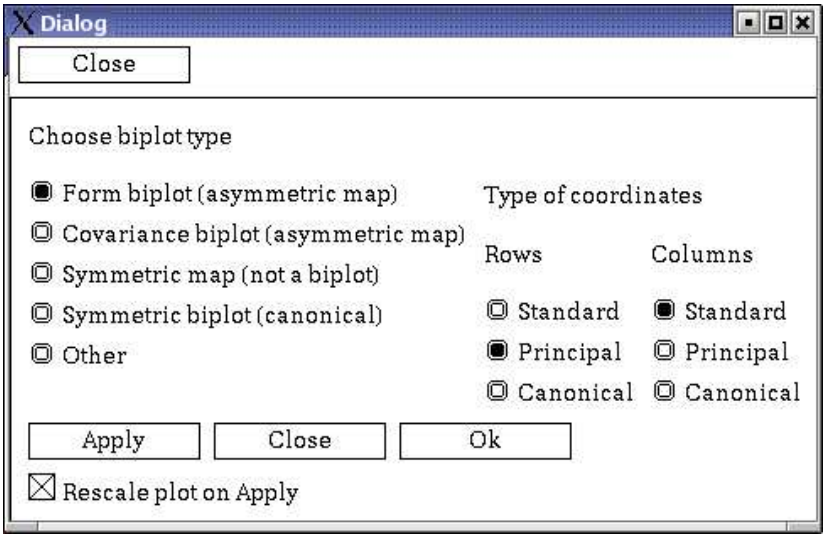

Figure 6: Dialog window used in XLS-Biplot to specify the biplot type.

4.2.3 Building the biplot

The type of biplot desired may be specified by the user using the dialog window shown in 6. The four standard options described in section 2.4 are available, and the user is free to choose any other combination of standard/principal/canonical coordinates.

Once the biplot has been shown, the user may reverse the direction of any of the two axes by using the dialog window shown in Figure 7, left.

The aspect ratio of the graph is always fixed to $1:1$ and the range of both axes is always the same, according to the discussion in section 2.6.

4.2.4 Interaction and edition

Once the biplot is in its window (see Figure 4) user interaction is mainly controlled by using the tools palette on the left part of the window. There are two buttons that open a menu and six buttons that modify the mouse mode.

The *Biplot...* menu give access to data windows where the original or the transformed data may be visualized. Using that menu, it is also possible to define or modify the transformation to be applied to original data (see Figure 5) or the biplot type to draw (see Figure 6)

The *Save...* menu provides items for saving both graphical and numerical output as described in the next two sections.

When the *Info* tool is in use, pointing to elements in the graph results in some information appearing in the message area of the window (see Figure 4). Clicking on an element provides more detailed information on that element (a point, an arrow,a label or the axis).

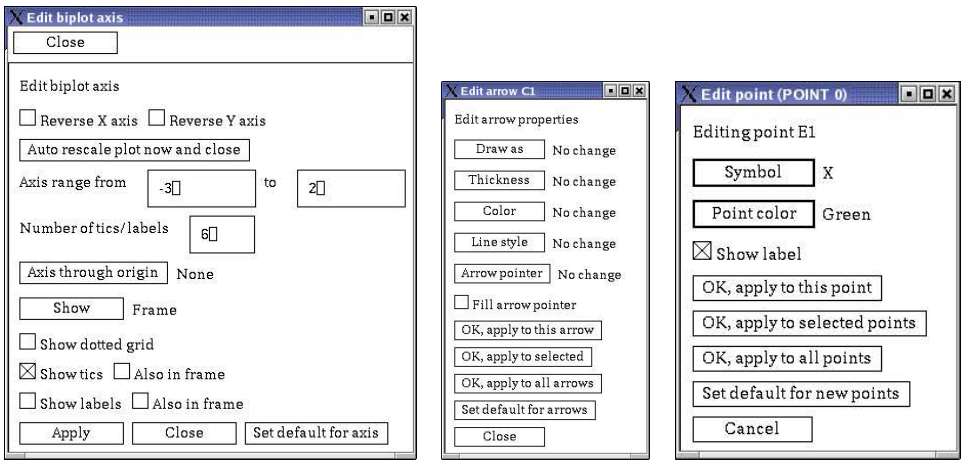

Figure 7: Dialog windows used in XLS-Biplot to customize (left to right) axis properties, arrows/axis representing variables and points representing cases.

When the *Edit* tool is in use, clicking on a point, an arrow/axis or on the axis of the graph gives access to dialog windows (see Figure 7) to modify the corresponding object. The desired modifications can be applied to a single or to several elements. The *Select* tool makes it possible to select several elements to be modified at once. This tool is also useful as a data analysis tool: selecting any element (case or variable) is reflected also in the data window (provided it is already open). Using these dialog windows it is also possible to set the current properties for axes/arrows/points as default for future biplots.

The *Move* tool allows the user to modify the length of the arrows keeping fixed the relative lengths to one another, or even to change the length of a single arrow. It is also possible to move the labels to place them in a better position than the one assigned automatically by the (a bit smart) default algorithm.

Finally the *Coords* tool is useful to show the coordinates of any click in the graph, and the *Help* tool gives access to a context sensitive series of help screens.

4.2.5 Numerical output

Numerical output is accessible through the item *Save biplot Report* in the *Save...* menu button (see Figure 8). The report can be directed to a text file or displayed in a window.

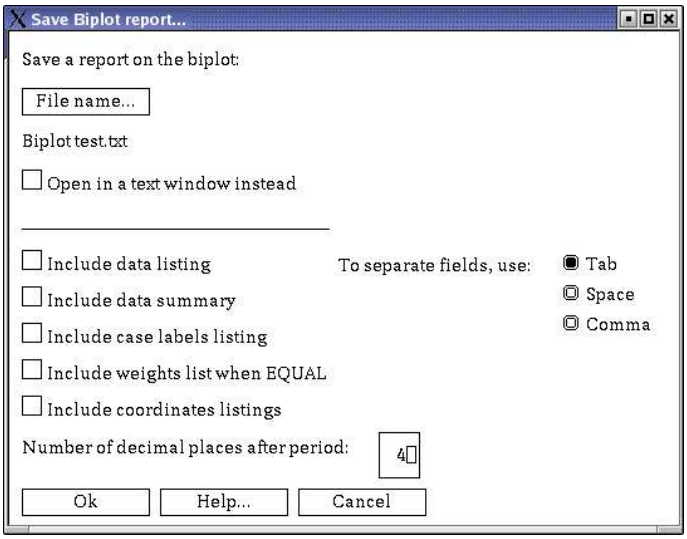

Figure 8: Dialog window used in XLS-Biplot to obtain a numerical report on the characteristics of the biplot and the data used to build it.

The report includes a brief summary of the data and the options used to build the biplot, and the eigenvalues obtained in the SVD and the subsequent percentage of variability captured in each dimension (see section 2) as well as the quality of the representation of the variables in the plot. Optionally, the biplot report may include the data list, a statistical data summary, case label listing, weight listing or transformed coordinates listing.

4.2.6 Graphical output

The screen-resolution (low quality) graph produced by XLS-Biplot in its window may be used in many ways depending on the operating system it runs on. Under Linux/Unix it is possible to save a postscript file containing the screen-resolution graph. Under MS-Windows, the graph may be printed or copied to the clipboard and then pasted to any MS-Window application.

The XLS-Biplot graph may be saved in a variety of formats, the main one being the `fig` format that can then be edited and converted by using `xfig` or `fig2dev`, see the Appendix or the user's manual for details. In this way high quality graphical output (postscript or pdf, for example) may be obtained (see Figures 2 and 3 obtained by this way).

4.3 The Biplot Server on the Web

We have set up an experimental *Biplot Web Server* that makes XLS-Biplot graphs available through the Internet by using any Browser. In its current state, the mechanism is very simple and limited: by using a standard web browser (see the Appendix) the user should be able to paste a simple data matrix (containing nothing but some lines of numbers delimited by tabs) and send it. A page will be sent back with the biplot and some links to obtain postscript or pdf versions of the graph.

4.4 Some technical details

In XLS-Biplot we make extensive use of the object-oriented programming system available in Xlisp-Stat. It is strongly needed to easy implementation of the interactive tools and it is also very convenient for efficient implementation of the computation flow. For this, we use the approach described in Udina (2000): the needed quantities and/or intermediate results are computed just once when they are need and stored until anything they depend on changes, and this is done by an automatic mechanism without the programmer care.

In Table 1, we list the object classes used in XLS-Biplot with a brief description of their functionality. More detail on how to use these objects and XLS-Biplot from Xlisp-Stat programs may be found in the user's manual, see the Appendix.

In Table 2 we list the different file formats involved in XLS-Biplot.

References

Bond, J. and G. Michailides (1997). Interactive correspondence analysis in a dynamic object-oriented environment. *Journal of Statistical Software 2*(8), 1–30.

Class (inherits from)	Description
`bp-proto` (`graph-proto`)	The main object class, graphical method in file `bp-window.lsp`, computational methods in `bp-proto.lsp`
`toolbar-proto` (`graph-overlay-proto`)	Display and manage the tool bar appearing in the biplot window. See the file `toolbar-ovl-proto.lsp`.
`toolbar-button-proto` (`*object*`)	The buttons that make up the toolbar. See the same file above.
`poor-text-window-proto` (`graph-proto`)	A window to show formatted text, with tabbed columns. It is needed because the `text-window-proto` is not available in all Xlisp-Statincarnations. Implemented in file `poor-text.lsp`
`data-show-proto` (`graph-window-proto`)	A spreadsheet-like window to show data tables. Data can not be edited, but cells, row or columns can be selected. It's linked to the biplot window.
`menu-button-proto` (`text-item-proto`)	An item for dialog windows that includes a pop-up menu and mechanisms for interaction.
`path-item-proto` (`menu-button-proto`)	A menu-button object specialized to select path to files.

Table 1: Object classes used in XLS-Biplot.

File format	Extensions	Description	Created by	Readable by
Text delimited	.dat, .tab	Data with spaces or tabs between data pieces	Any text processor, Excel, etc.	XLS-Biplot
Biplot format	.bpdata	Data with a special format (see 4.2.1)	XLS-Biplot	XLS-Biplot
Biplot report	.bpreport	A textual report of the Biplot	XLS-Biplot	Any text proc., Excel
Fig graphics	.fig	Graphics description for Xfig	XLS-Biplot, xfig, jfig	xfig, jfig, fig2dev (translation to ps)
Postscript	.ps .eps	Page description	fig2dev, etc.	Ghostscript, Gsview, pstoedit, LaTeX
Adobe Portable document format	.pdf	Page description	Acrobat, pstoedit, gsview	Acrobat Reader, gsview
MS-Windows metafile	.wmf .emf	Graphics description	pstoedit from ps under MS-Windows	MS-Office apps, etc.
Jpeg and Png	.jpg, .jpeg, .png	Compressed graphics, Portable Network Graphics	XLS-Biplot using `gs` or fig2dev	Web browser, Photo Editors, xv

Table 2: A summary of the different file formats involved in XLS-Biplot

Friendly, M. (2000). *Visualizing Categorical Data.* Cary, NC: SAS Institute Inc.

Gabriel, F. B. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika 58*, 453–467.

Gower, J. and D. Hand (1996). *Biplots.* Chapman and Hall, London.

Greenacre, M. (1993). *Correspondence Analysis in Practice.* Academic Press, London.

Lipkovich, I. and E. P. Smith (2002). Biplot and singular value decomposition macros for Excel©. *Journal of Statistical Software 7*(5), 1–15.

Tierney, L. (1990). *LISP-STAT, An Object-Oriented Environment for Statistical and Dynamic Graphics.* New York: Wiley.

Udina, F. (2000). Implementing interactive computing in an object-oriented environment. *Journal of Statistical Software 5*(3), 1–20.

A Finding the software and manual

The main site for XLS-Biplot is `http://tukey.upf.es/xls-biplot`. From there, it is possible to download the software for Unix and MS-Windows versions of Xlisp-Stat. The MS-Windows version includes an extended version of Xlisp-Stat made by Forrest Young and collaborators for ViSta that has some convenient extra utilities. The user's manual, also included in the mentioned distributions, is accessible at

`http://tukey.upf.es/xls-biplot/users-manual/`

The user's manual includes full directions to install and run the software.

The biplot web server is accessible at

`http://tukey.upf.es/bp-form.html`

in its quite limited current version. From user provided data, it provides a principal components biplot in several graphics format (png, ps, pdf).