A Tribute to J. Bertin’s Graphical Data Analysis

Antoine de Falguerolles
Laboratoire de Statistique et Probabilités, UMR CNRS
Université Paul Sabatier
118 route de Narbonne
F 31062 Toulouse Cedex

Felix Friedrich & Günther Sawitzki
StatLab Heidelberg
Im Neuenheimer Feld 294
D 69120 Heidelberg

Summary

Bertin's permutation matrices give simple and effective tools for the graphical analysis of data matrices or tables. We discuss some abstractions which help understanding Bertin's strategies and can be used in an interactive system.

1. Introduction

In [Bertin 1977], J. Bertin introduced a display and an analysis strategy for multivariate data with low or medium sample size. Bertin tries to make the information in a data set understandable. He does not fit models: he tries to provide simple tools to interrogate data. The tools operate simultaneously on cases and variables, combining aspects otherwise separately encountered in cluster analysis (on cases) and principal component analysis or factor analysis (on variables). Bertin's approach has received attention over the years in various areas [see Caraux 1984 for references]. It has been introduced to the SoftStat audience by P. Kremser in 1987, followed by various subsequent contributions. In the general fields of statistical data analysis and data visualization however it has received only marginal resonance. Bertin's work deserves more attention. We give a short introduction to one aspect, Bertin's permutation matrices.

Bertin formulated his ideas on the background of the technical facilities of the seventies. New possibilities have become available meanwhile. We have to read Bertin's work with these changes in mind. Instead of taking his proposals as verbatim recipes, we have to look for the underlying ideas and concepts, and translate them to the possibilities we have now. Our way to do this is to formulate Bertin's strategies in an abstract way, translating them into actions in a second step.

2. Bertin Matrices

In abstract terms, a Bertin matrix is a matrix of displays. Bertin matrices allow rearrangements to transform an initial matrix to a more homogeneous structure. The rearrangements are row or column permutations, and groupings of rows or columns. To fix ideas, think of a data matrix, variable by case, with real valued variables. For each variable, draw a bar chart of variable value by case. Highlight all bars representing a value above some sample threshold for that variable.

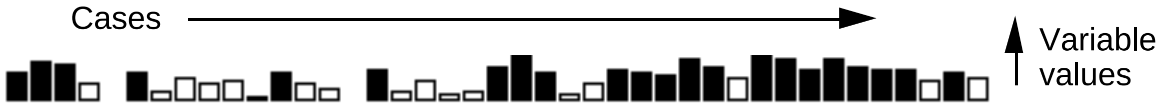


Figure 1. Univariate simple Bertin plot of one quantitative variable.

Arrange the bar charts as rows of a matrix. As an example, we use the results of 9 referenda for the 41 Irish communities [http: see below] (two values missing). Four of the referenda refer to abortion. One of these ("Right to Life") is formulated in the terms preferred by the government and appears as a near inverse of all other formulations. Fig. 2 shows the raw data matrix.

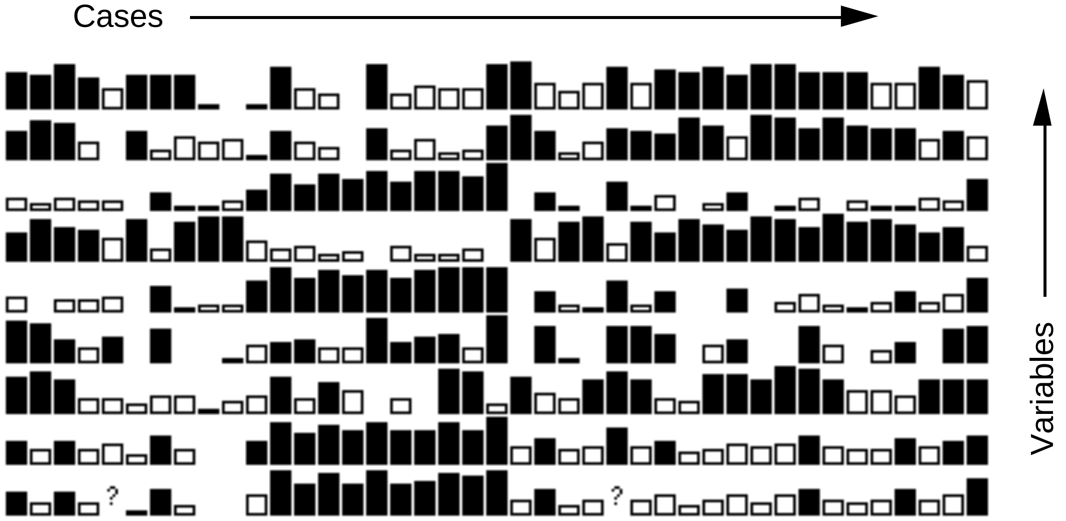


Figure 2. Raw Bertin matrix of the Irish data. Each row shows the results of one referendum, each column one case (a community). Values exceeding the variable mean are highlighted black.

Row or column permutations can obviously clean up this matrix. A few permutations, selected ad hoc by geometric considerations, lead to Fig. 3, with apparent groups separated by cut lines.

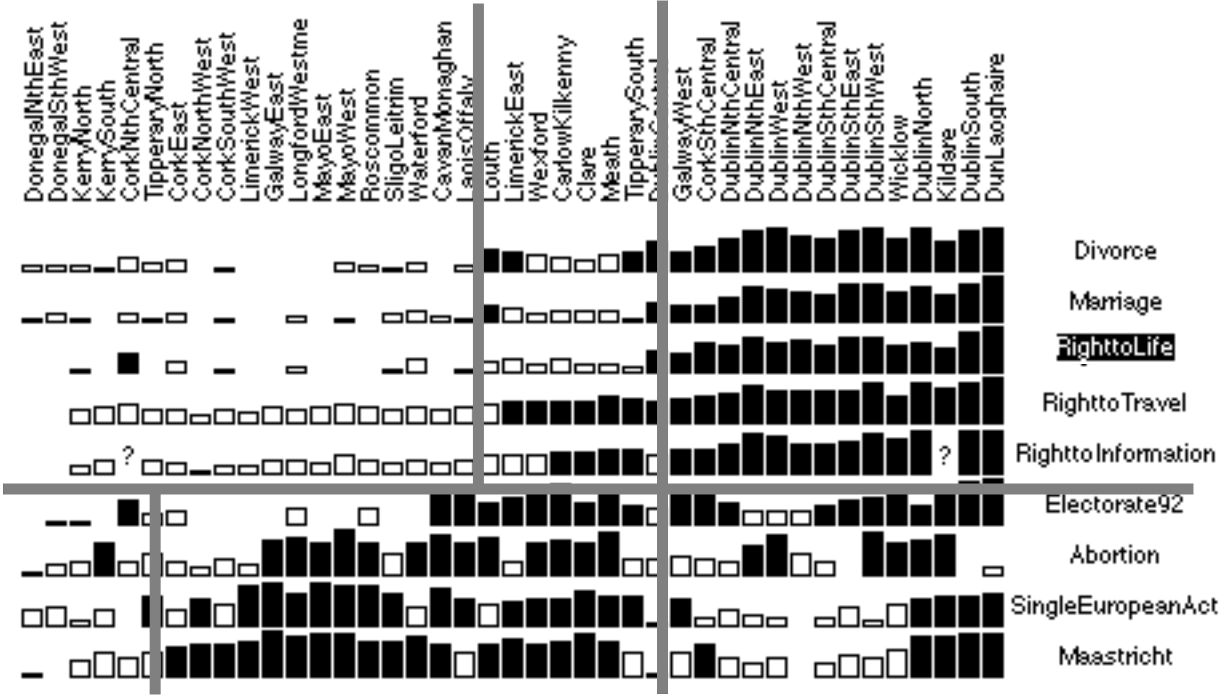
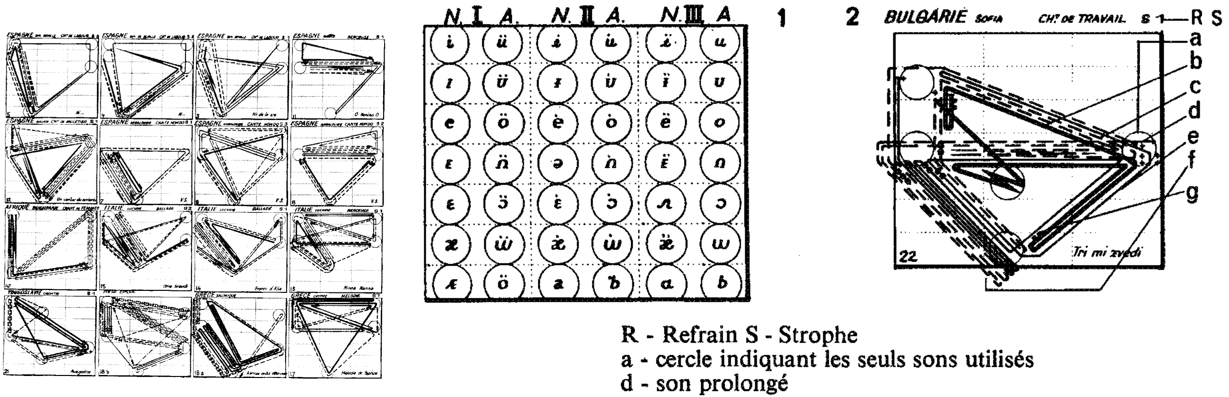


Figure 3. Bertin matrix of the Irish referenda data: rearranged, and apparent split lines added. Data for "Right to Life" used in complemented form.

The labels had not been used to define the arrangement. It show that referenda on the European unity have been separated from all other referenda, as is the group of industrial areas (Dublin, Cork,

…). Even using background it would be hard to find a better structuring than Fig. 3. Of course the arrangement is still disputable. Should Dublin Central be included in the top right group, or in the next group left? Should the “Single European Act” and the Maastricht referendum be isolated as a special group? Some entries seem to fit nowhere, like the results for Cork Northern Central.

The first step is constructing the elementary displays. To allow rapid visual inspection, the elementary displays must enhance semantic contrasts and serializations. This is a topic of its own. We focus on the second step, rearrangement, which comprises ordering(sorting) and grouping. Rearrangement takes the bulk of the work. The elementary displays are constructed only once. The arrangement is done repeatedly, possibly using various strategies and objectives. The starting point need not be a raw data matrix. For example, it may be the concatenation of two way tables having one classifying factor in common. The individual displays in a Bertin matrix are not restricted to bars or some other representation of quantitative information. To illustrate the scope of Bertin matrices, Fig. 4 (left) shows an excerpt of a more complicated graph.



R - Refrain S - Strophe
a - cercle indiquant les seuls sons utilisés
d - son prolongé
b, c, e, f - 1re, 2e, 3e, 4e phrases
g - même son répété deux fois de suite

Figure 4. Left: Bertin matrix of the Folk Song Data Set. Each display gives the vowel sequence in one verse of a song. Exhibits 1 and 2 show the construction rule of the individual displays (excerpt of the raw plot [Bertin 1977, p. 126]).

3. Formal Bertin Matrices

For a formal definition, we start with a data matrix X_0 (K variables by N cases). Based on X_0, we have a matrix of displays. The display cell for observation i and variable j may depend on non-local information derived from X_0. In the Irish data example given above, it needs highlighting information defined by $X_0[i,j]>\mathrm{Mean}(X_0[.,j]))$. For simplicity, we assume that X_0 has been augmented by additional components if necessary and we identify X_0 with the initial graph. Operating on the initial graph, we have a group of permutations Π. Unless structural restrictions apply, Π is the full product group of row and column permutations $\Pi=\Pi_{row}*\Pi_{column}$. We use X for matrices(graphs) after permutation, so $X=\pi X_0$ for some permutation π, and $X[i,j]=\pi X_0[i,j] = X_0[\pi^{-1}(i,j)]$.

To formalize Bertin’s strategy we introduce a purity function $\Phi=\Phi(X)$ that measures the simplicity of the Bertin plot. A formal aim could be to find Φ -optimal arrangements, that is permutations π^* maximizing $\Phi(\pi X_0)$. This does not yet cover the grouping of variables or cases. For the moment, we will leave this aspect open and hide it in a proper definition of the purity function Φ. Of course the formalization still falls short of real applications. In any application, we will not have just one purity function, but a family of objectives and strategies. Nevertheless the idea of a simple optimization problem can help.

Often a purity function will compare the entry for a cell with neighbouring entries. A side problem is that variables may have different scales. Comparison between cases may be easy, but comparison across variables needs precautions. The usual considerations for scaling from multivariate analysis apply. We will use ranking by variable as a preferred scaling. Since there may be missing data, we have to standardize and use relative ranks. For short, we use the term “ranks” for relative ranks by variables with Rk[i,j] the relative rank of X[i,j]. We do not rank by case. Up to discretization this rank scaling transforms our variables to a uniform distribution on [0,1]. Using some inverse of the distribution function leads back from [0,1] to the original variable scaling.

Purity functions can be of arbitrary complexity, but the most common ones are constructed from simpler elements. To fix ideas, we give some examples.

- Sorting: given rows or column scores, define the purity as the number of pairs in “correct” order. Arranging for this purity function means sorting by score.
- Arranging: given row or column distances, define the purity as the sum of distances of adjacent rows/columns. Although this is a simple idea, the complexity is threatening:
- Smoothing: assume a local smoothing. Use the negative sum of norms of residuals to define a purity.

For sorting, we expect a complexity of O(M log M), where M is O(K+N). The second example gives a warning. If we select a pivot element, arrangement is a sorting problem. If we allow free arrangement, finding a Φ -optimal (or good) arrangement π^* is a travelling salesman problem. The third example opens the door to really hard problems.

Any metric on permutation groups can be used to construct a purity function. Candidates are discussed in [Diaconis 1988, Ch.6B]. For applications, we try to restrict to simple variants of the problem. If a purity function can be decomposed for $= (_{row}, _{column})$ as $\Phi() = c(\Phi_{row} (_{row}), \Phi_{column}(\pi_{column}))$ for some connecting function c, we can hope to reduce the general optimization problem to separate arrangement problems for rows and columns. As an example, assume the rank transformation defined above and define row and column distances as d(X[i,.],X[j,.]) = $\sum_k$ | Rk[i,k],Rk[j,k] | resp. d(X[.,i],X[.,j]) = $\sum_k$ | Rk[k,i],Rk[k,j] |. The row distances are Spearman’s footages. The column distances are a slight modification, taking into account that we did rank within variables, not cases. Since the row (column) distances are invariant with respect to column (row) permutations, the arrangement problem can be decomposed into row and column arrangement.. An optimal arrangement for the Irish data using this distance is shown in Fig. 5.

The role of formal optimization should be seen clearly. In most real applications, a proper purity function or metric comes with the ultimate model and is one of the results of an analysis, not the beginning. During the modelling phase, actions should be kept as flexible as possible. Partial steps sometimes may be formulated in full right as optimization problem, and we should use formal optimization at its best to pass these steps.

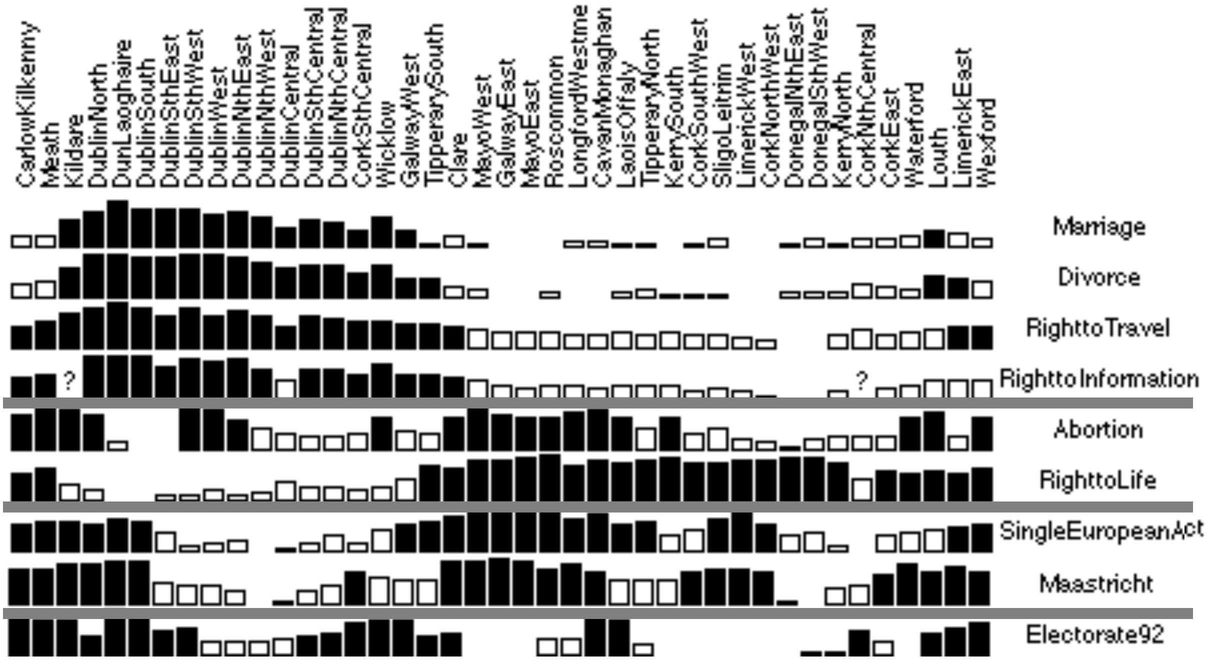


Figure 5. Optimal arrangement for the Irish data for row and column L_1-distance on ranked variables. Split lines mark close groups, based on the permutation distribution of this distance.

It is interesting to see the differences between the ad hoc solution Fig. 3 and Fig. 5. Fig. 3 reflects sorting processes, pushing variables to extremes, while Fig. 5 uses the freedom to arrange variables and cases smoothly on two sides. As in Fig. 3, we see patches with a homogeneous structure in Fig. 5 on the left and middle, and not all variables are homogeneous in the same way. We find a group of “hard to fit” communities to the right of the arrangement – starting with the previously noticed Cork Northern Central.

As with all statistical procedures, a critical question is whether π^* reveals any true information, or whether it is but an expression of random artefacts. Given any purity function, one test for non-randomness is obvious: let $\Phi^* = \Phi(\pi_o X_o)$ be a maximal value of $\Phi(\pi X_o)$. An effect is significant at level α if $\#\{\pi: \Phi(\pi(X_o)) \geq \Phi^*\} / \#\Pi \ \leq \ \alpha$.

4. Basic Bertin Actions

> *Ce point est fondamental. C’est la mobilité interne de l’image qui charactérise la graphique moderne. [Bertin 1977, p. 5]*

Bertin has designed his strategy for interactive use, first for manual sorting of marked cards. In the meantime, better interactive facilities have become available on computers. As has been pointed out by Bertin, there are certain aspects of his strategy that can be easily automated, like initial preparation of the data, sorting and arranging according to some specified optimality criterion. There are other aspects, such as the proper formulation of hypothesis, implicit modelling, and usage of lateral information that need direct interaction. Combining both in an interactive environment meets technological boundaries. The challenge for an interactive system is to find a sufficient repertoire of actions that spans the full group of operations while each single operation has an immediate clear interpretation and allows an efficient combination to improve the purity of the graph.

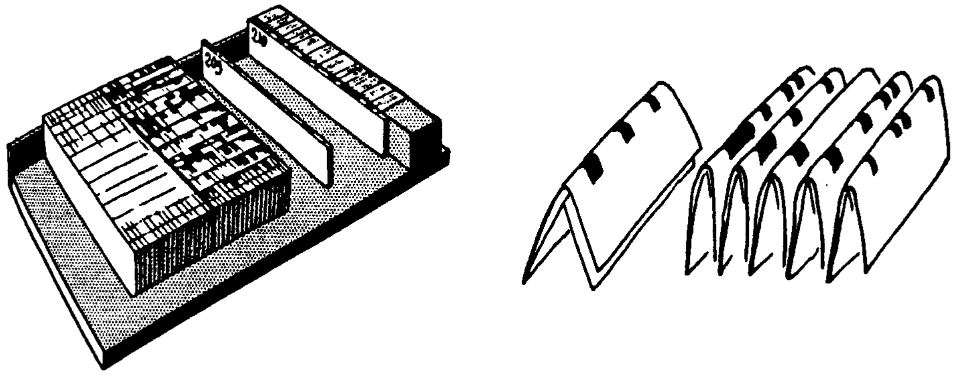


Figure 6. Two mechanical realizations of Bertin arrangements [Bertin 1977 p.71].

Some actions are obvious. Usually the permutation groups are represented using swaps as elementary generators. In an interactive environment, this is an irritating action, giving the impression that two rows or columns are changed simultaneously. For an interactive environment, a more stable impression is given by using shift operators as generators.

Shift(i,i') move i to new location after i'.

Shifts are needed for rows and columns. We omit this remark for the actions further on. In a mechanical system, shift actions can be easily extended to operate on consecutive rows or columns. On a computer, we can even use non consecutive selections. For any sequence of variables or cases S we have a shift operation Shift(S,i'). The exact action is: rows $S=\{i_1,..\}$ resp. columns $S=\{i_1,..\}$ move consecutively after row i' resp. column i' in their previous order. This allows to use the parallelism is inherent in optical perception. The second class of actions is grouping of variables, implemented for example by inserting a separator. A split level count allows hierarchical grouping. We make life still easier for us by introducing a split operator defined in terms of a list of rows and columns of the current graph Γ. We identify these lists with the group S defined by this split,

Split(S) define row/column selection S of Γ as a new group.

Shifting and splitting generate the full set of action to go from some starting display to any Bertin rearrangement, and are all actions needed for manual intervention. These actions allow for an immediate interpretation in an interactive environment. If they are integrated in an automatic arrangement system , a score should be supported which reflects the necessary actions to align cases or variables, such as Ulam's distance which counts the number of necessary shifts [Gordon 1979].

All actions discussed here and below are implemented in the Voyager system [Sawitzki 1996]. As a convenience, sorting is added, using a stable sort algorithm to allow a predictable combination of sorting passes. Additional transformations are included such as transformations to ranks, normalized and studentized values, pairwise correlations, footage distances etc.

5. Bertin Strategies

> *Et que constate-t-on alors? C'est que la simplification n'est que le regroupement de ce qui se ressemble. L'œil simplifie à condition de pouvoir corriger, en permutnat les lignes, les irrégularités qu'il constate dans le désordre initial. L'inventaire est un désordre, introduit par les aleas de l'imagination et les contingences des catégorisations générales. L'œil simplifie. [Bertin 1977, p. 7]*

Elementary actions are sufficient for small data sets or simple purity functions, in particular if we are prepared for interactive work. For larger data sets, the mere number of individual actions and inspections soon will keep us from achieving reasonable results in realistic time. Time can be reduced with partitionable strategies. Looking at Fig. 3 or Fig. 5, we recognize patches with homogeneous patterns. For example in Fig. 3, the top five variables have a strong positive correlation and have very small correlation with the variables below the split line. Information like this can be used to partition the problem. The strategy suggested by Bertin is: take a pivot variable, and sort the rest in order of correlation. Separate the high correlation group from the around zero group and high negative correlation. As a next step, each of these groups can be analyzed as a separate matrix.

There are special cases which are worth special attention. In the original rearrangement of the Irish data, the “Right to Life” occurred as a single referendum with extremely negative correlation to the others at top of Fig. 3. This was the reason to use it in complementary form, fitting smoothly with the top group. The general strategy is to assign topological information with group boundaries. A group of cases or variables may be accepted as well separated, and discrepancies across the boundary can be ignored in the purity calculation. Or a group may be considered “complementable”, allowing permutations “over the edge” removing a variable from the low correlation set and inserting its negative values in the high correlation set or vice versa. To allow these actions in an automatic arrangement scheme, we augment the information with the boundary structure. For nearest neighbour sums, for example, we modify the definition of the purity function appropriately to take into account an adapted definition of neighbourhood.

6. Patches

> *Mais tout change si le dessin est matériallement reclassable. En effet, la perception visuelle est spatiale, et permet à quiconque d’utiliser un nouveau principe de reclassement: a la prise en compte simultanée de plusieurs elements. [Bertin 1977, p. 7]*

In section 4, we assumed that we can split cases or variables into groups. But Fig. 3 illustrated that sometimes a split should not apply to all cases or variables - we need recursive splits on any split component defined so far. Another restriction to give up is to think of split components as disjoint rectangles. Usually they will be at the time we define them. But using additional actions or changing the strategy will spread them over a fabric. In an interactive environment, or for theoretical analysis, we can abandon these restrictions. We can concentrate on the idea behind these split components. And we can support additional attributes. In the previous section, we have seen the need to associate attributes to a split component, like a classification of the boundary as “separating” or “complementable”. The general idea is that we have some homogeneity within a split component. This may mean that we have a strong consistency, like in the split component on the top right of Fig. 5. For the cases in this component, virtually any variable in this component can represent any other, and we could well replace the information in this part by any single surrogate variable. On the contrary, the bottom right component is a drop box, but by variables it seems to be clearly separated from the variables to its top, as well by cases to the left. We would note it for a second analysis, and classify it as a “noisy” part for now.

For a formal definition, the first step is to define the support of a split component. It need not be rectangular, but we still suppose that it has been defined as a rectangle in some arrangement. We call it a patch: formally, a patch is defined by a list ((i_1,..), (j_1,..)) of variable resp. case indices. We always use these indices in terms of the original variable resp. case indices. The permutation at the time of the definition of a patch is one of its attributes. So with a patch, we have permutations π_{row}, π_{column} defining the native neighbours for elements in a patch. A patch may be classified as “noisy” and in this case we will use it at best to define some selection of cases and variables. Or it may be classified as homogeneous. In this case, we assume some local model which can be used to replace the patch information by some “local smoother” - a local surrogate, preferably of lower complexity. Patches may be overlapping. The collection of patches in use will be denoted by *S*, and we allow the purity function to depend on *S*.

Patches are familiar with recursive partitioning approaches, like CART. We go beyond this and do not require patches to be disjoint, thus moving from recursive partitioning to recursive coverage. Using the general Bertin approach, we allow to define patches both in terms of cases and variables, thus going beyond other related approaches.

7. Extended Bertin Actions

L’information est la résponse à une question. [Bertin 1977, p. 11]

The elementary actions are extended to use patches. Shift and Split actions may be defined by patches. A Shift moves a patch to the bottom right of a cell, keeping (or restoring) the native order of the patch. A Split marks the boundaries of a patch. Patches are intended for groups with homogeneous internal structure. Three actions, Expand and Collapse, and Substitute are used to go from the original information to a placeholder. Collapse hides the patch, ignoring its structure in purity calculations. Substitute replaces the patch by (preferably simpler) surrogate information and Expand restores the “original” state. Collapsing is a way to allow a graphical representation if screen space is sparse, or if distracting information should be moved out of the way.

Sorting comes in two variants. A sort restriction only sorts cases and variables within a patch. Sorting by patch information on the other hand sorts the full data set, based on information about the patch S. Both sortings have an automatic grouping associated to them. Sorting can be done stepwise for all variables (or cases) of a patch, respecting the highlight attribute. This leads to a recursive partitioning.

While sorting has a well defined target and uses pairwise comparisons, the general arrangement leads to the full complexity of an arrangement problem. Patches may be used to reduce the complexity to a feasible amount. Like sorting, the general arrangement can be restricted to a patch, or defined by a patch.

8. Relation to other methods

Bertin refers to V. Eliseeff’s “scalogram” [Eliseeff 1968, 1970] as source of the matrix rearrangement idea. Interactive modifications, as used with Bertin matrices, are common today. Bertin matrices are particular in that they are in no way restricted to simple data structures (such as quantitative real valued data). And they are particular in the scope of the actions, allowing to access and rearrange information both by case and by variable. Cases and variables are treated differently in that the summary information on variables is usually needed for the elementary plots. For example, scaling and highlighting will depend on some per variable statistics. For the arrangement, both are treated the same way. Free permutations are allowed, unless structural restrictions apply.

There is an obvious relation to several other approaches. Bertin matrices can be viewed as special parallel coordinate plots [Inselberg 1996]. While usual parallel coordinate plots us variables for ordinates, Bertin matrices conventionally operate on the transposed matrix using cases for ordinates. Bertin plots operate both on cases and variables and try to recover structural information. In this aspect, they are related to techniques like the biplot [Gabriel, 1971]. In the simple case where patches give a (hierarchical) partition, the cuts of a Bertin rearrangement can be formulated as a graphical model. For more complicated cases, where patches may form a general covering and where case and variable structuring aspects enter simultaneously, the range of graphical models will soon be too restrictive.

Bertin matrices can be used as a visualization tool. Benzécri suggests that the scores of order one provided by a correspondence analysis be used for a ranking of rows and columns in a Bertin matrix [Benzécri 1973, see also Hill 1974]. Correspondence analysis is a particular bilinear model [Ducros et al. 1997]. The latter can be applied to analyse more complex data sets than a two way contingency table, thus providing a ranking for a Bertin matrix. For the example of the Irish referenda data, a bilinear model (with independent Gaussian error and logit link) gives a ranking as in Fig. 7. The “Electorate 92” has not been used in the model but is inserted for reference only.

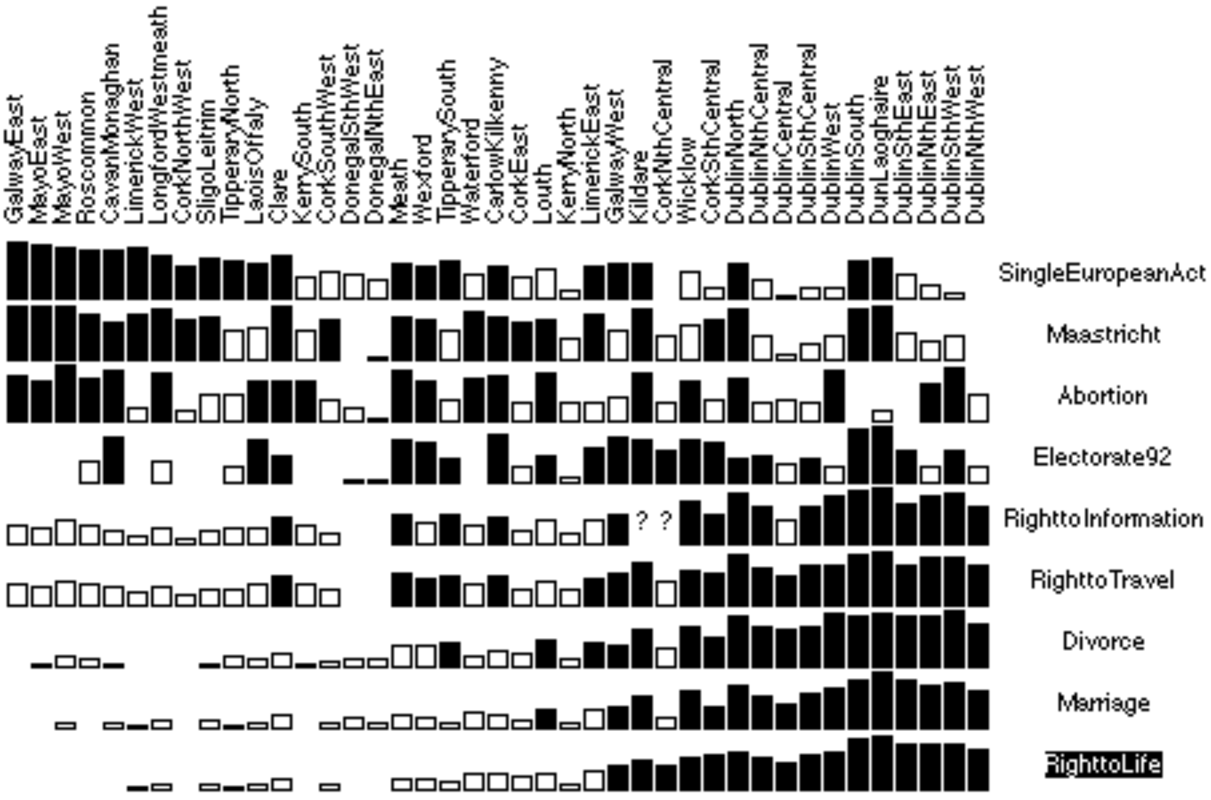


Figure 7. Optimal arrangement of the Irish referenda data using bilinear scores.

The arrangement problem has been addressed in Trellis [Becker et al. 1996]. The most important difference is that Trellis aims to give a reliable and predictable strategy for arrangement based on information about the scales, ranges and order relations of the variables, leaving identification of correlations and interactions as a second step, whereas the Bertin approach uses correlations and interactions first to define the arrangement. Bertin matrices add interactive aspects.

The usage of patches is related to recursive partitioning [Breiman et al. 1984] or recursive covering approaches [Friedman 1996]. Instead of defining a coverage of the variable space, we are using a coverage of the sample space. This allows more flexibility while keeping most advantages of the DART approach. The introduction of surrogate variables and the use of standardized ranks in their definition has been stimulated by [Tukey 1991]. Again the idea to concentrate on subsets of the observations is specific to the Bertin approach. We concentrated on the “mechanics” of Bertin arrangements. A proper statistical analysis, can be done using permutation approaches, as has been indicated in the discussion of the null distribution. The necessary tools can be found in [Diaconis 1988].

Benzécri, J.-P. (1973). *L’analyse des données*. Vol. 2. Dunod: Paris.

Bertin, J. (1977). *La graphique et le traitement graphique de l’information*. Flammarion: Paris.

Breiman, L., Friedman, J.H., Olshen, R., Stone, C.J.(1984). *Classification and regression trees*. Wadsworth.

Caraux, G. (1984). Reorganisation et representation visuelle d’une matrice de donnees numeriques: un algorithme iteratif. *Revue de Statistique Appliqué*, 32, 5-23.

Diaconis, P. (1988). *Group representations in probability and statistics*. Hayward: California.

Ducros, D.; Mondot, A.-M.; de Falguerolles, A. (1997). AIRLS and IRALS algorithms for generalized bilinear models. *These proceedings*. To appear.

Eliseeff, V. (1968). Application des propriétés du scalogramm à l’étude des objects.*Journeées d’Études sur les Méthodes de Calculs dans les Sciences de l`Homme*. CNRS: Paris.

Eliseeff, V. (1970). Archéologie et calculateurs. *Problèmes sémiologiques et mathématiques*. CNRS:Marseille.

Friedman, J.H. (1996). Local learning based on recursive covering. Preprint. Stanford <ftp://playfair.stanford.edu/pub/friedman/dart.ps.Z>

Gabriel, R.K. (1971). The biplot-graphic display of matrices with application to principal component analysis. *Biometrika*, 54, 453-467.

Gordon, A.D. (1979). A measure of the agreement between rankings. *Biometrika*, 66, 7-15.

Hill, M.O. (1974) Correspondence Analysis: a neglected multivariate method. *Applied Statistics*, 23, 340-350.

Inselberg, A. (1996). Parallel coordinates: a guide for the perplexed. Manuscript. Tel Aviv.

Sawitzki, G. (1996). Extensible statistical software: On a voyage to Oberon. *Journal of Computational and Graphical Statistics*, 5, 263-283.

Becker, R.A., Cleveland, W.S., & Shyu, M.-J. (1996). The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, 5, 123-156.

Tukey, J.W. (1991).Use of many covariates in clinical trials. *International Statistical Review*, 59, 123-137.

Irish referenda data: Communicated by A. Unwin et al.
<http://www1.Math.Uni-Augsburg.DE/~theus/Workshop/Data.html>

The Bertin matrices have been produced using Voyager [Sawitzki 1996]. We thank L. Dümbgen and G. Reinelt for helful comments, and T. Christof for calculating the exact optimal placement for the Irish referenda data.

Contact:

<mailto:gs@statlab.uni-heidelberg.de>
<http://www.statlab.uni-heidelberg.de/projects/bertin/>