Geometric Representation of Multivariate Data Frames

Jan de Leeuw

We discuss two classes of drawing methods for multivariate categorical data. Both are inspired by multidimensional scaling, and are intimately linked to the notion that similarity in the data is naturally represented as distance in a low-dimensional Euclidean space. The objects that are measured, or categorized, by our variables are represented as points. Each variable defines a partition of the points into subsets corresponding with the values of the variable.

The first class of methods are the clumping methods, that try to represent the objects with the same values on a variable by small compact subsets of space. Since there are many ways to measure the size of a point set, there are many clumping methods. The second class are separation methods, which try to construct smooth surfaces from some parametric family to separate points having different values on the variable.

Clumping and separation methods can be implemented using either least squares or likelihood based algorithms, which define the two main ways to measure and minimize badness-of-fit.

1. Data

Data are n observations on m categorical variables. Variable j takes k_j different values, where $1 \leq k_j \leq n$. Numerical variables are just a special case.

There are two basic ideas in the techniques we discuss. The n objects are represented as points in low-dimensional Euclidean space in such a way that objects with similar *profiles* (values on the variables) are relatively close (in some sense).

Somewhat more specifically, if two objects have the same value on a variable, then that should make them more close, if they have the same values on all variables, then they should map into the same point.

One easy way to portray the results of an analysis is to make m copies of the plot of the n objects and label them by the values (categories) of the variable.

2. Clumping

In this class of techniques we use a measure of the size of a cloud of points in $\mathbb{R}^p$. Suppose X are the coordinates of the points representing the n objects. Variable j defines a partition into k_j subsets, which have measures

$$\sigma_{j1}(X), \cdots, \sigma_{j,k_j}(X).$$

We also define a measure $\sigma(X)$, which measures the size of the cloud of all n points.

It is now easy to define a loss function by the simple rule

$$\lambda(X) = \frac{\sum_{j=1}^{m} \sum_{\ell=1}^{k_j} \sigma_{j\ell}(X)}{\sigma(X)},$$

although other ways of combining the size measures into a single numerical loss value are certainly possible.

A clumping technique finds n points $x_i \in \mathbb{R}^p$ such that $\lambda(X)$ is minimized.

Many measures of size have been proposed, and some have actually been studied from a computational points of view.

We mention the edge-length of the minimal spanning tree, the circumference of the convex hull, the radius of the circular or elliptical hull, the sum of the distances to the Weber point, the maximum distance between two points.

Anybody is free to suggest their own measures of homogeneity. That's the easy part. However, implementating good algorithms is quite another matter.

We discuss one particular choice of size in more detail. Each point set with k points has a *star*, which is the set of lines connecting the k points with their centroid (average). The size of the star is the sum of squares of the k lines.

The loss function associated with this definition of size defines *multiple correspondence analysis* (or *homogeneity analysis*). It is closely related (but mathematically far simpler) than the *Weber correspondence analysis* we mentioned before.

This can be expressed in matrix notation, which already suggests why this is a fortunate choice.

Let G be the n x k binary *indicator matrix* (a.k.a. *dummy*) indicating which objects belong to which categories. Also $D = G'G$. Then $Y=D^{-1}G'X$ are the k centroids, and

$$\sigma(X) = \mathbf{tr}\ (X - GY)'(X - GY)$$

is the sum of the sizes (squared line lengths) of the k stars.

The clumping technique we derive from measuring star size (with squared distances) minimizes $\sigma_1(X) + \cdots \sigma_m(X)$ over all centered configurations X such that $\mathbf{tr}\ X'X = 1$. This clumping technique is due to Louis Guttman in the early forties.

The optimization problem turns out to be an eigenvalue problem for the matrix

$$P_\star = \frac{1}{m}\sum_{j=1}^{m} P_j,$$

where

$$P_j = G_j D_j^{-1} G_j'.$$

Alternatively we can define

$$G = (G_1|\cdots|G_m).$$

And solve the generalized eigenvalue problem $Cy = m\lambda Dy$, where $C=G'G$ and $D=\mathbf{diag}(C)$.

In fact, what I just said is not quite correct. We have to require $X'X=I$ instead of $\mathbf{tr}(X'X)=1$. Then X corresponds to the eigenvectors corresponding to the p largest eigenvalues (except for the largest one, which is always 1 and corresponds to a column of ones).

The resulting technique (*MCA* or *homals*) is computationally very efficient. In the Gifi (1990) system MCA is generalized to minimizing

$$\sigma(X; Y_1, \cdots, Y_m) =$$

$$= \frac{1}{m}\sum_{j=1}^{m} \mathbf{tr}\ (X - G_jY_j)'M_j(X - G_jY_j)$$

over both X and Y with various restrictions on the Y.

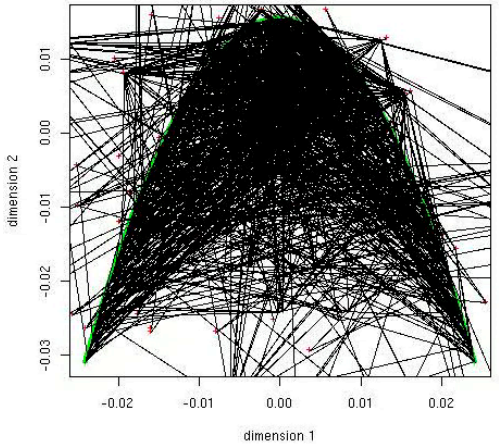
Some of the other clumping techniques (size measures) have been tried, and they generally have failed.

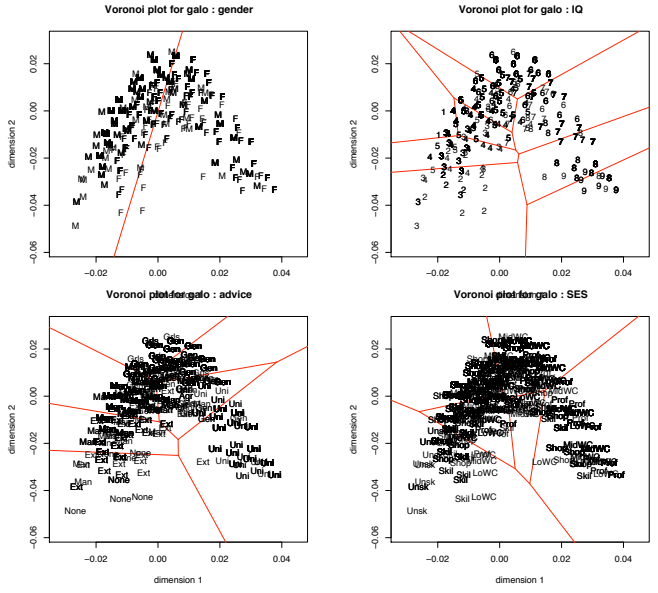
Two problems with MCA is that it is somewhat inelegant that we have to use a *normalization* to force multidimensional solutions.

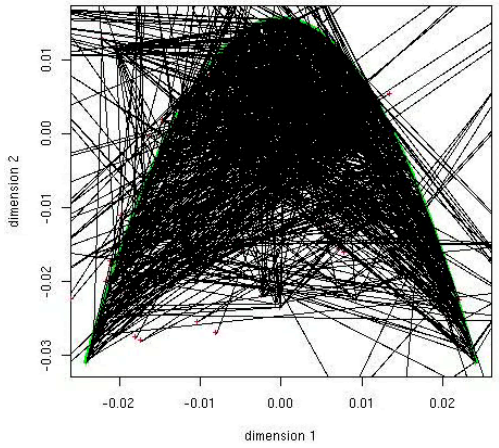
Moreover requiring orthogonality produces *horseshoes* (for well-understood reasons). Subsequent dimensions are quadratic, cubic, ... functions of the first dimension.

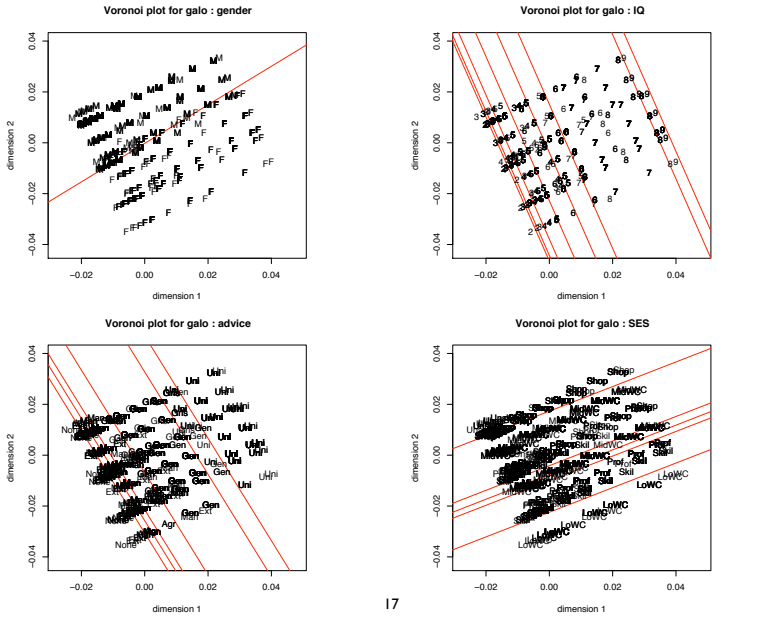
To some extent we can get around this by restricting the Y_j, for instance requiring them to be of *rank one*. This encourages the centroids to be on straight lines through the origin.

The GALO Data

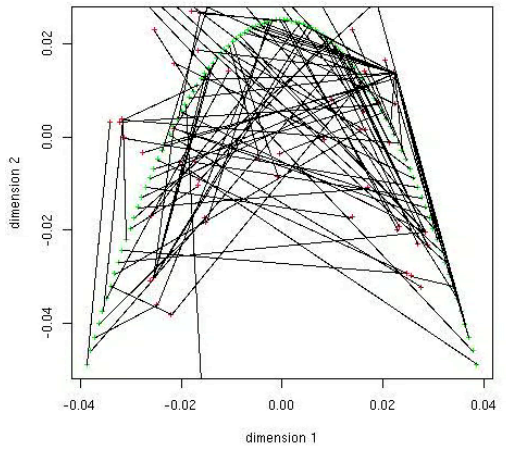


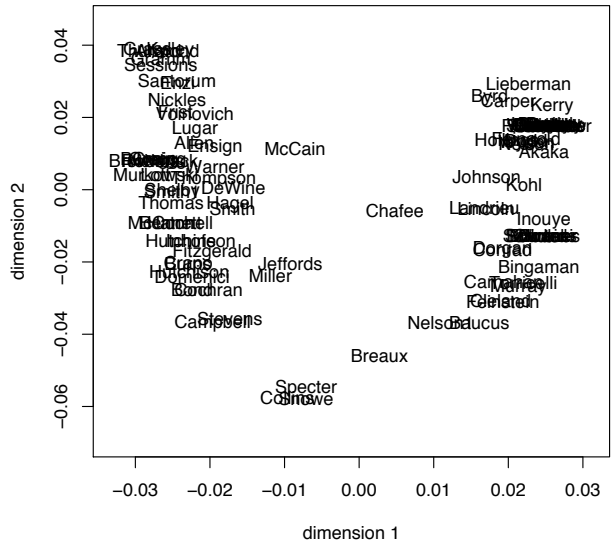






The Senate Data





3. Separation

A different starting point is not to concentrate on making certain sets of object points small in size, but on locating the points in space in such a way that they can be *separated* by simple parametric curves or surfaces.

Suppose, for instance, we have a variable with k ordered categories (values). We want to locate the n object points in, say, the plane such that the k subsets can be separated by parallel lines, i.e. the subsets must be in parallel strips.

Or, alternatively, suppose we have a binary variable taking values "aye" and "nay". We want to locate the n points x_i in the plane such that there is a circle with all the "aye" points in the circle and all the "nay" points outside the circle.

Separation techniques have been around since the mid-sixties, and they are direct descendants of the Shepard-Kruskal approach to non-metric multidimensional scaling (1965). Although there is Guttman (1940) and Coombs (1950). Because of this ancestry the original separation techniques mostly use least squares loss functions.

We will go another way here, because we want to measure fit (loss) on the probability scale. This dispenses with the need for normalization, and it may get rid of the horseshoes.

Consider the loss function

$$\Delta(X,Y) = \\ = -\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{\ell=1}^{k_j} g_{ij\ell}\log\frac{\exp(-\|x_i-y_{j\ell}\|^2)}{\sum_{\nu=1}^{k_j}\exp(-\|x_i-y_{j\nu}\|^2)}.$$

Let's interpret optimizing this loss as a geometric problem. If we can find X and Y such that

$$g_{ij\nu} = 1 \Leftrightarrow \|x_i - y_{j\nu}\| < \|x_i - y_{j\ell}\| \qquad \forall \ell \neq \nu,$$

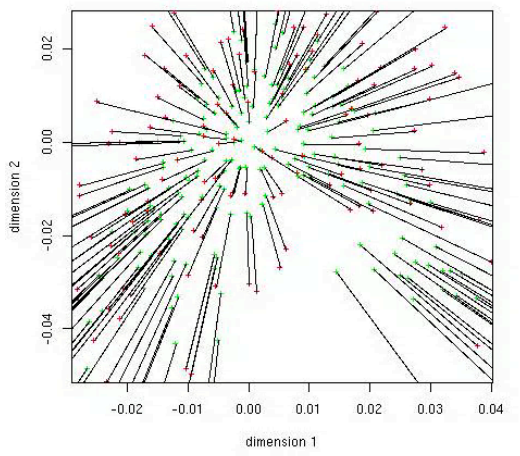
or, in words, such that each object is closest to the category the object is in, then we can get arbitrarily close to a solution with loss equal to zero.

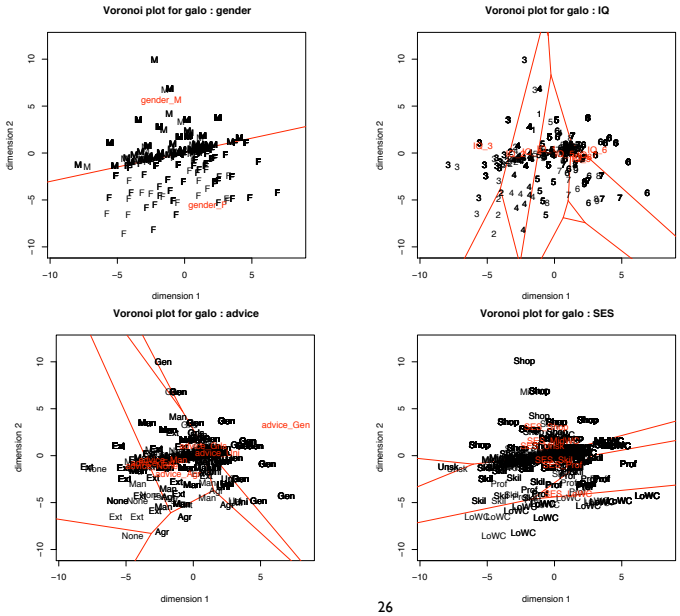
So one way to think of the method is to find an approximate solution to a system of nonlinear inequalities (Motzkin, Agmon, 1950)

Where are the separations ? Well, if we let the categories of each variable define Voronoi cells in $\mathbb{R}^p$, then the inequalities say that objects should be in the *correct* Voronoi cells (for all variables). Voronoi cells provide a non-parametric system of separations.

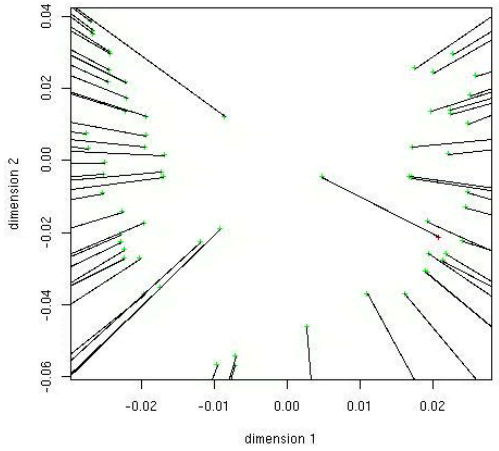
As with MCA we can make this parametric by constraining the Y_j. Requiring them to be of rank one, for instance makes the Voronoi cells into strips bounded by parallel lines or planes.

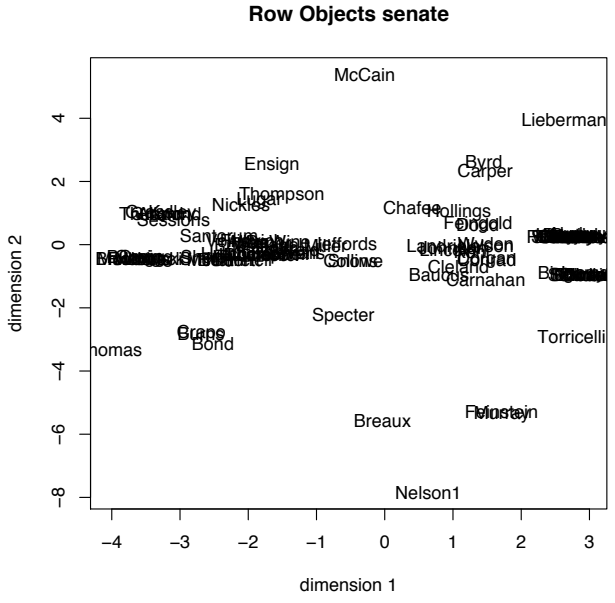
GALO Data





The Senate Data





Conclusions

Separation techniques became computationally feasible only very recently. They use complicated majorization algorithms.

They provide an interesting alternative to the much more familiar (much less expensive, and much better understood) least squares techniques.

http://www.cuddyvalley.org/psychoR