Gifi Goes Logistic

SCASA Keynote 2005-11-05

Jan de Leeuw

The *Gifi System* is one particular way to define and organize a large class of descriptive multivariate analysis techniques. The system allows the techniques to be implemented in in a modular series of computer programs.

A. Gifi, *Nonlinear Multivariate Analysis*, Wiley, 1990.

G. Michailides and J. de Leeuw, *The Gifi System of Descriptive Multivariate Analysis*, Statistical Science, 13, 1998, 307-336.

We start with an *n x m* data matrix *X*, with *n observations* on *m variables*.

This is, of course, the basic format for data in packages like SPSS and SAS, in spreadsheet programs such as Excel, and (as the *data frame*) in R.

For the time being, we assume (*without loss of generality*) all variables are discrete, and variable *j* has k_j *categories*.

A *Data Matrix*

a	p	u
b	q	v
a	r	v
a	p	u
b	p	v
c	p	v
a	p	u
a	p	v
c	p	v
a	p	v

can also be coded as

a	p	u	3
a	p	v	2
a	q	u	0
a	q	v	0
a	r	u	0
a	r	v	1
b	p	u	0
b	p	v	1
b	q	u	0
b	q	v	1
b	r	u	0
b	r	v	0
c	p	u	0
c	p	v	1
c	q	u	0
c	q	v	0
c	r	u	0
c	r	v	0

Profile Frequencies

Multidimensional Cross Table

$$\left[\begin{array}{ccc} (a) & u & v \\ p & 3 & 2 \\ q & 0 & 0 \\ r & 0 & 1 \end{array}\right] \left[\begin{array}{ccc} (b) & u & v \\ p & 0 & 1 \\ q & 0 & 1 \\ r & 0 & 0 \end{array}\right] \left[\begin{array}{ccc} (c) & u & v \\ p & 0 & 1 \\ q & 0 & 0 \\ r & 0 & 0 \end{array}\right]$$

Indicator Matrices and Indicator Super-matrix

	a	b	c	p	q	r	u	v
01	1	0	0	1	0	0	1	0
02	0	1	0	0	1	0	0	1
03	1	0	0	0	0	1	0	1
04	1	0	0	1	0	0	1	0
05	0	1	0	1	0	0	0	1
06	0	0	1	1	0	0	0	1
07	1	0	0	1	0	0	1	0
08	1	0	0	1	0	0	0	1
09	0	0	1	1	0	0	0	1
10	1	0	0	1	0	0	0	1

To indicate the generality of indicator matrix (dummy) coding, we discuss some extensions that can be used. This also shows how so-called continuous variables can be handled

-- missing data
-- fuzzy coding
-- interactive coding
-- spline coding

Missing Data

$$\begin{bmatrix} a \\ b \\ a \\ <NA> \\ b \\ c \\ a \\ a \\ c \\ <NA> \end{bmatrix} \underset{\text{single}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \underset{\text{multiple}}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} \underset{\text{deleted}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}$$

Fuzzy Coding

1	a	1	0	0
2	b	0	1	0
3	<NA>	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
4	a	1	0	0
5	b	0	1	0
6	c	0	0	1
7	a	1	0	0
8	a	1	0	0
9	c	0	0	1
10	<NA>	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

Smearing

		0	1	2	3	4	5	6
1	3	0	0	0.25	0.50	0.25	0	0
2	2	0	0.25	0.50	0.25	0	0	0
3	3	0	0	0.25	0.50	0.25	0	0
4	5	0	0	0	0	0.25	0.50	0.25
5	1	0.25	0.50	0.25	0	0	0	0
6	1	0.25	0.50	0.25	0	0	0	0
7	4	0	0	0	0.25	0.50	0.25	0
8	4	0	0	0	0.25	0.50	0.25	0
9	3	0	0	0.25	0.50	0.25	0	0
10	2	0	0.25	0.50	0.25	0	0	0

B-spline (order 2)

	[-1,1]	[0,2]	[1,3]	[2,4]	[3,5]	[4,6]	[5,7]
3.1	0.0	0.0	0.0	0.9	0.1	0.0	0.0
2.5	0.0	0.0	0.5	0.5	0.0	0.0	0.0
0.2	0.8	0.2	0.0	0.0	0.0	0.0	0.0
5.1	0.0	0.0	0.0	0.0	0.0	0.9	0.1
1.3	0.0	0.7	0.3	0.0	0.0	0.0	0.0
0.9	0.1	0.9	0.0	0.0	0.0	0.0	0.0
4.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
4.1	0.0	0.0	0.0	0.0	0.9	0.1	0.0
3.5	0.0	0.0	0.0	0.5	0.5	0.0	0.0
2.9	0.0	0.0	0.1	0.9	0.0	0.0	0.0

Interactive Coding

			ap	aq	ar	bp	bq	br	cp	cq	cr
1	a	p	1	0	0	0	0	0	0	0	0
2	b	q	0	0	0	0	1	0	0	0	0
3	a	r	0	0	1	0	0	0	0	0	0
4	a	p	1	0	0	0	0	0	0	0	0
5	b	p	0	0	0	1	0	0	0	0	0
6	c	p	0	0	0	0	0	0	1	0	0
7	a	p	1	0	0	0	0	0	0	0	0
8	a	p	1	0	0	0	0	0	0	0	0
9	c	p	0	0	0	0	0	0	1	0	0
10	a	p	1	0	0	0	0	0	0	0	0

The basis of the Gifi system for MCA is the notion of *homogeneity*, applied to a *joint plot* of objects and categories. This means we like the plot if:

- Objects with similar profiles are close together.
- Categories with similar content are close together.
- Objects are close to the categories they are in.

We now have to construct a quantitative measure of homogeneity, or rather of loss of homogeneity, which we will then minimize. Here are the basic ingredients.

- An $n \times p$ matrix X of *object scores*.
- $k_j \times p$ matrices Y_j of *category quantifications*.
- $n \times k_j$ indicator matrices G_j.

The Y_j can be collected in a $K \times p$ matrix Y, and the G_j in an $n \times K$ matrix G.

Now suppose we represent all n objects and all K categories as points in p-space. The coordinates are the rows of X and Y.

We can think of the indicator super-matrix G as the adjacency matrix of a graph, in which object i is adjacent to category k (of variable j) if i is in category k of variable j.

We can draw this graph by connecting the object points to all category points that they are in, so m lines depart from each object, for a total of nm lines in the drawing. This is the *graphplot*.

- The lines must be short.
- Normalize the object scores by $X'X = I$.
- Use squared distances to measure length.
- So we minimize the (squared) amount of ink in the graphplot.

More precisely, we must minimize

$$\sigma(X, Y) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{SSQ}(X - G_jY_j)$$

over both Y and the normalized X.

The core of the algorithm is the *reciprocal averaging* (or *alternating least squares*) algorithm. Start with some $X^{(0)}$. Then iteratively apply the *centroid principles*.

$$Y^{(k)} = D^{-1}G'X^{(k)},$$

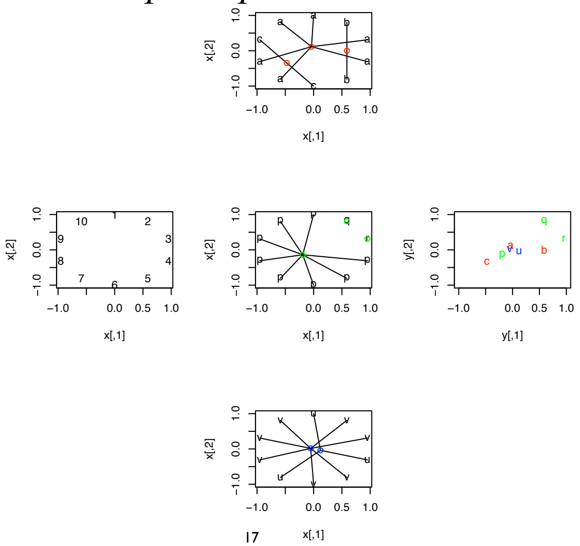
$$\tilde{X} = m^{-1}GY^{(k)},$$

$$X^{(k+1)} = \mathbf{orth}(\tilde{X}^{(k)}).$$

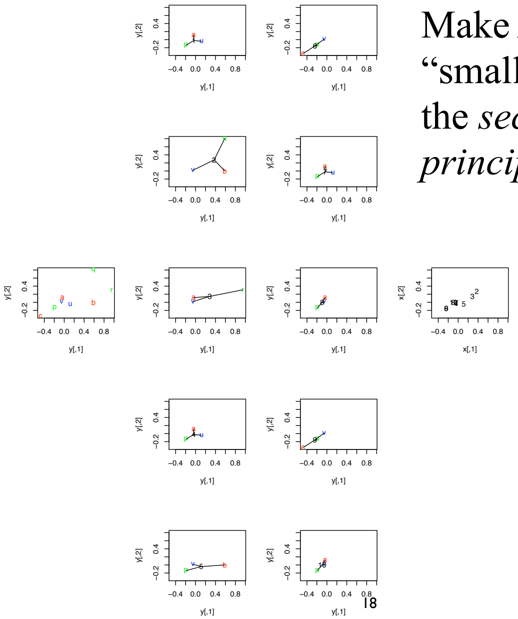
Here **orth**() is an *orthogonalizer*, such as Gram-Schmidt or QR or Procrustus.

Iterate until convergence, that is until $X^{(k)}$ no longer changes. Let us illustrate a single iteration.

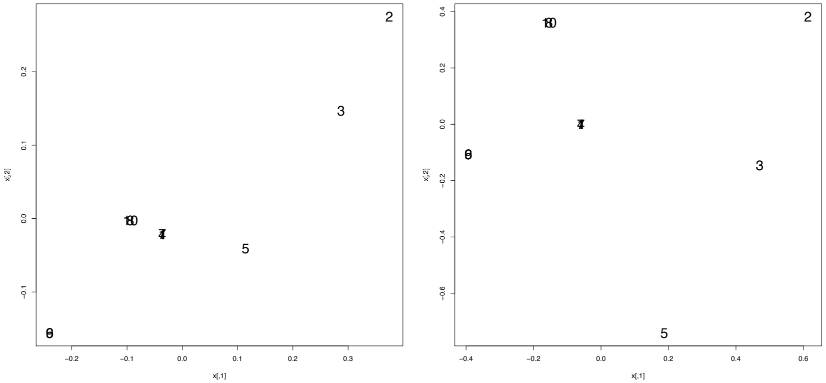
Make *Y* from *X* in the “small” example, using the *first centroid principle*. Start with a circle.



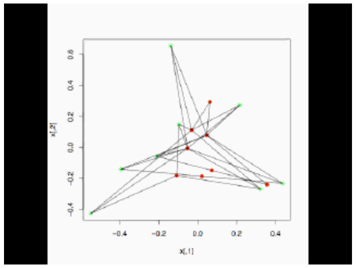
Make *X* from *Y* in the “small” example, using the *second centroid principle*.

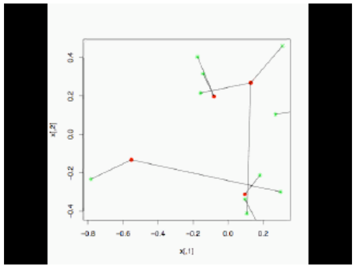


Finally: Normalizing X, using Gram-Schmidt.



We can illustrate the iterations using movies.





“Small” example

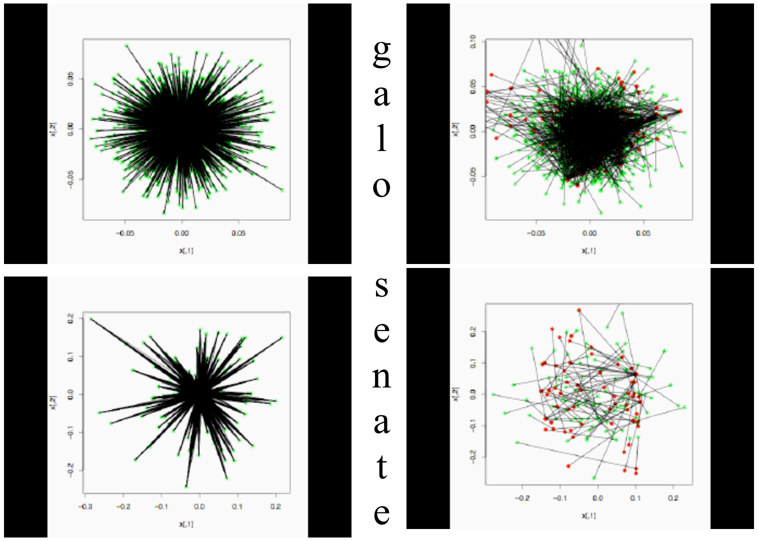
On the left: changes in the graphplot over iterations

On the right: changes in the object scores over iterations

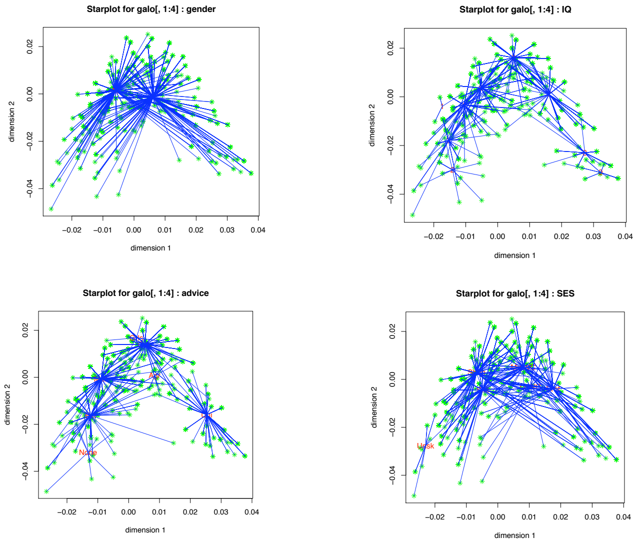
Some real examples we use throughout are

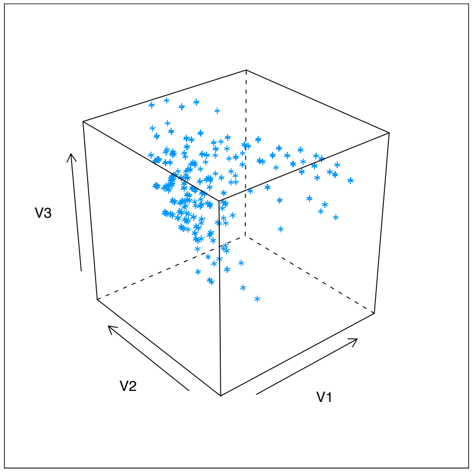
-- the GALO data (1290 students, gender (2), IQ (9), Advice (6), SES(7)).

-- the senate data, 20 votes in the 2002 US Senate (50 senators).

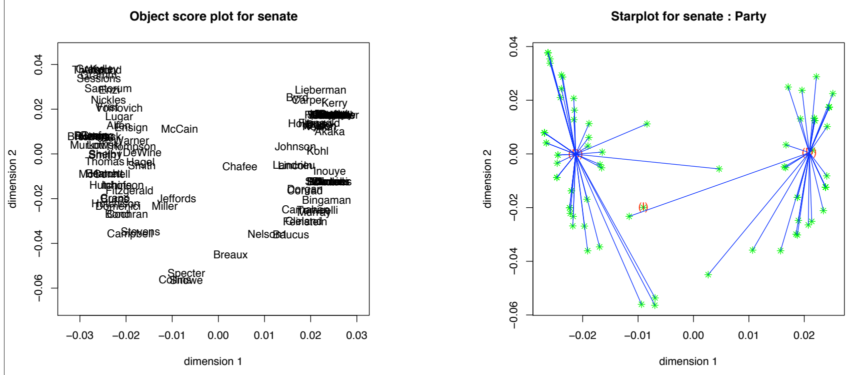


Starplots for GALO





GALO 3d object scores (cloud function from lattice package)



Senate with dédoublement and “party” passive

We can relate homogeneity to principal component analysis by restricting the category quantifications of all variables to be *on straight lines through the origin*. A separate line for each variable, of course. In other words, we require the category quantifications to be of rank one. In formula

$$Y_j = z_j a_j'.$$

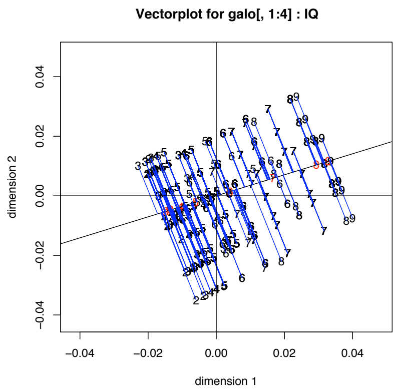
The z_j are called the *lower rank quantifications*, the a_j are the *category loadings*. They are written, by default, to the output file of the *homals* program.

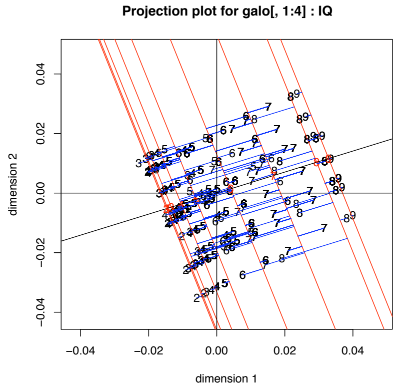
The algorithm using rank restriction is based on the partitioning

$$\sigma(X,Z,A) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{SSQ}(X - G_j\hat{Y}_j) +$$

$$+\frac{1}{m}\sum_{j=1}^{m} \mathbf{tr}(\hat{Y}_j - z_j a_j')' D_j(\hat{Y}_j - z_j a_j').$$

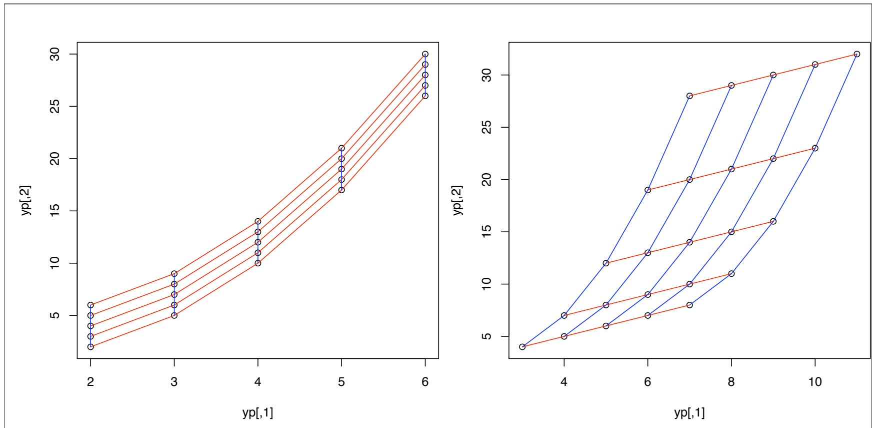
We use alternating least squares. The update for X given Y is the same as before. If we update Y, we use the rank restrictions, except for multiple variables.





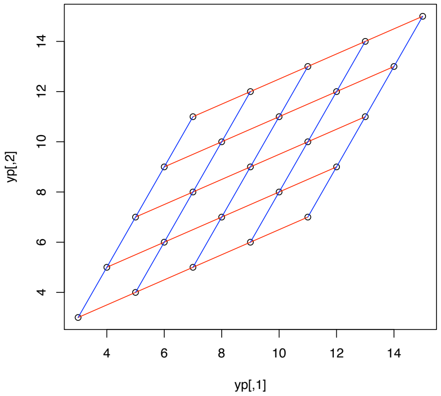
Regression and canonical analysis can also be incorporated in a simple way.We have seen that we can introduce the notion of sets of variables using *interactive coding*. All variables in a set become a single variable, and we then apply ordinary homogeneity analysis. But interactive coding can soon become impractical. Then we use *additivity restrictions*.

By using suitable partitionings of the variables into sets, and by using measurement and rank restrictions, we can construct generalizations of many classical descriptive multivariate analysis techniques.



$$y_1 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix} \qquad y_2 = \begin{bmatrix} 1 & 1 \\ 2 & 4 \\ 3 & 9 \\ 4 & 16 \\ 5 & 25 \end{bmatrix} \qquad y_1 = \begin{bmatrix} 1 & 1 \\ 2 & 4 \\ 3 & 9 \\ 4 & 16 \\ 5 & 25 \end{bmatrix} \qquad y_2 = \begin{bmatrix} 6 & 7 \\ 5 & 6 \\ 4 & 5 \\ 3 & 4 \\ 2 & 3 \end{bmatrix}$$

With both rank and additivity constraints, category quantifications are on a regular lattice grid.



$$y_1 = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \\ 4 & 8 \\ 5 & 10 \end{bmatrix} \qquad y_2 = \begin{bmatrix} 10 & 5 \\ 8 & 4 \\ 6 & 3 \\ 4 & 2 \\ 2 & 1 \end{bmatrix}$$

The Gifi loss function, incorporating additivity by using sets of variables, becomes

$$\sigma(X, Y) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{SSQ}(X - \sum_{\ell \in L_j} G_\ell Y_\ell).$$

Here the indicators can be fuzzy or incomplete, and there can be rank constraints on the category quantifications.

Software

The *homals* package in R does principal component analysis, correspondence analysis, multiple correspondence analysis, regression, canonical correlation analysis, and multiset canonical correlation analysis. It allows for treating variables nominal, ordinal, numerical; as well as single and multiple.

A similar set of options is available in *SPSS Categories*, except that they are distributed over various programs.

So far, we have measured homogeneity using Euclidean geometry, centroids, and correlations, and all loss functions lead to some form of *least squares*. We now depart somewhat from the framework and measure loss on a probability scale. This has the additional advantage that no arbitrary normalizations are needed.

Our basic data are still the indicator matrices, but our metric becomes *logistic likelihood*. This is not necessarily *better* than least squares, but it can be expected to lead to quite *different* solutions.

Let us look at a single indicator matrix *G* with *n* rows and *k* columns. The probability that object *i* is in category *l* is

$$\pi_{i\ell}(X,Y) = \frac{\exp(\phi(x_i,y_\ell))}{\sum_{j=1}^k \exp(\phi(x_i,y_j))},$$

for some suitable function ϕ, which we do not specify yet.

Assuming independence, we find for the *deviance* (-2 times the log likelihood)

$$\Delta(X,Y) = -2\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell=1}^{k_j} g_{ij\ell}\log\pi_{ij\ell}(X,Y).$$

This means that if object *i* is in category *l* of variable *j*, then we want

$$\phi(x_i,y_{jl}) > \phi(x_i,y_{j\nu}) \qquad \forall\nu.$$

If ϕ is homogeneous, i.e. if

$$\phi(\alpha x_i,\alpha y_{jl}) = \alpha^r\phi(x_i,y_{jl}).$$

for some non-negative r, and these inequalities are satisfied, then we can actually make sure that

$$\pi_{ij\ell}(X,Y) \Rightarrow 1$$

and the deviance can be made equal to zero.

For instance

$$\phi_{ij\ell}(X,Y) = -\|x_i - y_{jl}\|.$$

This means we wants objects to be closer to the category they are in than to any other category, i.e. objects need to be in the *Voronoi cell* of the category.

Or

$$\phi_{ij\ell}(X,Y) = x_i'y_{j\ell} + \alpha_{j\ell}$$

which means categories are separated by hyperplanes, which can be chosen to be parallel by using rank one restrictions.

Instead of least squares we use *majorization* to arrive at a sequence of related least squares problems.

The general idea behind majorization, in the form in which we use it here, supposes we minimize a function with an upper bound for the Hessian. So $x'\mathcal{D}^2 f(y)x \leq Kx'x$. This implies

$$f(x) \leq f(y) + (x-y)'\mathcal{D}f(y) + \frac{1}{2}K(x-y)'(x-y),$$

and gives the convergent algorithm

$$x^{k+1} = x^k - K^{-1}\mathcal{D}f(x^k).$$

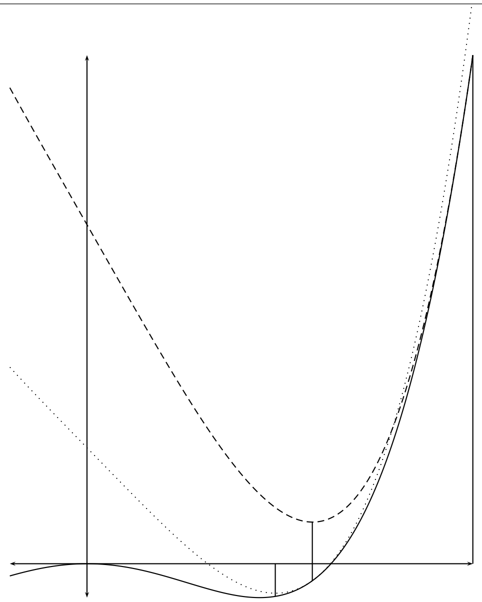
For a constrained problem the least squares sequence becomes

$$x^{k+1} \in \underset{x\in X}{\mathbf{argmin}}(x-z^k)'(x-z^k),$$

with

$$z^k = x^k - \frac{1}{K}\mathcal{D}f(x^k).$$

Now apply this to our logistic likelihood.



$$\frac{\partial \log \pi_{ijl}}{\partial \phi_{ij\nu}} = \delta^{\ell\nu} - \pi_{ij\nu},$$

$$\frac{\partial^2 \log \pi_{ijl}}{\partial \phi_{ij\nu}\partial \phi_{ij\omega}} = -(\pi_{ij\nu}\delta^{\nu\omega} - \pi_{ij\nu}\pi_{ij\omega}),$$

It follows that

$$-\sum_{\nu=1}^{k_j}\sum_{\omega=1}^{k_j}\frac{\partial^2 \log \pi_{ijl}}{\partial \phi_{ij\nu}\partial \phi_{ij\omega}}x_\nu x_\omega \leq \frac{1}{2}\sum_{\nu=1}^{k_j}x_\nu^2,$$

and this bound can be used to majorize the deviance.

After a great deal of calculation we see that in each iteration we must minimize

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell=1}^{k_j}(\phi_{ij\ell}(X,Y) - \tilde{z}_{ijl})^2,$$

where

$$\tilde{z}_{ij\ell} = \phi_{ij\ell}(\tilde{X},\tilde{Y}) + 2(g_{ij\ell} - \pi_{ij\ell}(\tilde{X},\tilde{Y})).$$

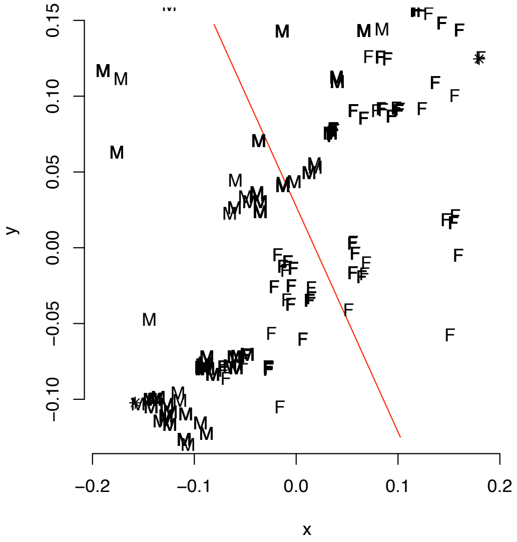
Thus (a) we are back to least squares, and (b) no normalization is needed.

In the case of negative distance we must minimize

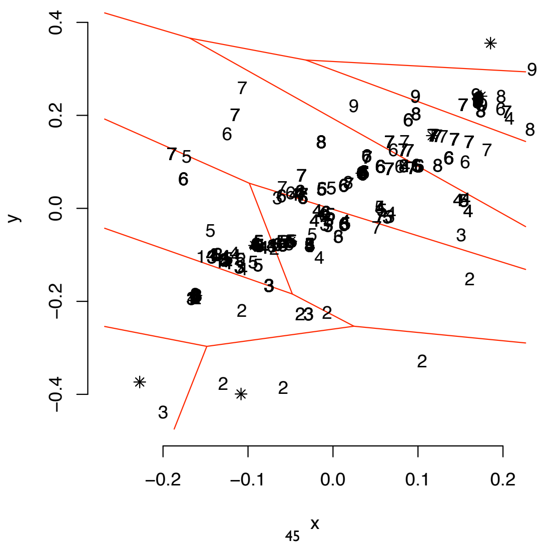
$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell=1}^{k_j}(d(x_i, y_{j\ell}) - (-\tilde{z}_{ij\ell}))^2$$

This is a metric unfolding problem (with the additional complication that the target values or dissimilarities may be negative). We can use existing unfolding algorithm to update the object scores and category quantifications.

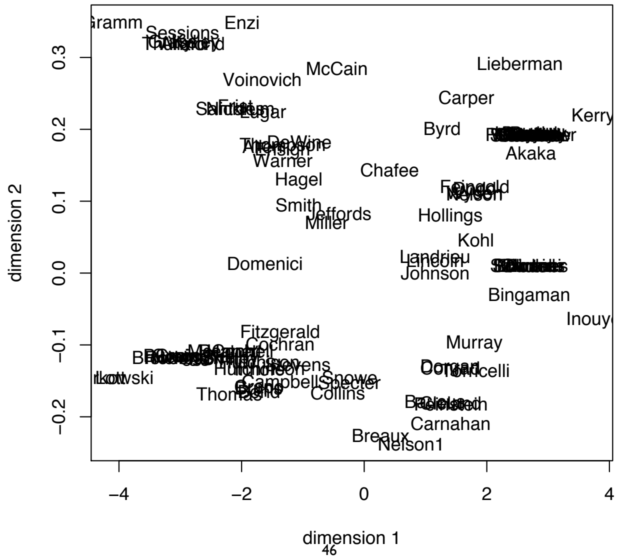
GALO: Gender

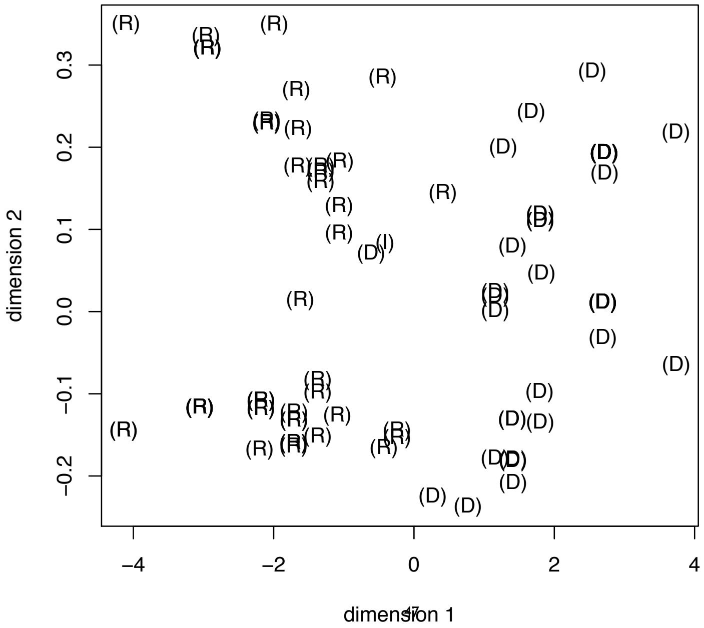


GALO: IQ



Senate Voting





The algorithms (written in R) are still very tentative, especially compared with the least squares methods which are very well tested and well-understood.

Principal component analysis of binary data by iterated singular value decomposition
Computational Statistics & Data Analysis, Volume 50, 2006, 21-39