Majorization Algorithms for Logit, Probit, and Tobit Models

Jan de Leeuw

Majorization Algorithms for Logit, Probit, and Tobit Models

Abstract

For a large variety of discrete choice models (or contingency table models) efficient and stable maximum likelihood methods can be constructed based on the majorization method. The course introduces majorization methods for algorithm construction. We show how to use the majorization principle to reduce complicated optimization problems to sequences of weighted or unweighted least squares problems.

Majorization methods are then applied to data analysis techniques used in economics, political science, psychometrics, ecology, sociology, and education.

Part I: Minimizing Loss

Many problems in computational statistics are, or can be cast as, optimization problems that maximize a numerical goodness-of-fit function or minimize a loss function.

Such problems are often solved by using general purpose optimization routines based on as steepest descent, conjugate gradient, or Newton methods. General purpose methods tend to work well for relatively small problems, but often need to be tweaked for large problems with many parameters.

So let's make the problem more specific and make some assumptions along the way.

We are given a continuous non-negative loss function $\phi : \Omega \to \mathbb{R}^+$. Our problem is to compute $\inf_{x \in \Omega} \phi(x)$ and, if the minimum exists, the place where it is attained.

This covers maximum likelihood, least squares, minimum chi-square, and so on.

We will study some general classes of iterative algorithms to solve this problem.

Iterative algorithms are described by *algorithmic maps*

$$A : \Omega \to \Omega$$

which compute *iterative sequences* by

$$x^{(k+1)} = A(x^{(k)}).$$

The behavior of such sequences is described by two key theorems. The first theorem addresses (global) convergence, the second theorem (local) rate of convergence.

Theorem (Zangwill). Suppose the map $A : \Omega \to \Omega$ is continuous and satisfies $\phi(A(x)) < \phi(x)$ for all $x \neq A(x)$. Then $\phi(x^{(k)})$ converges to, say, ϕ_∞. For any subsequence $x^{(\ell)}$ converging to, say, x_∞, we have $x_\infty = A(x_\infty)$ and $\phi(x_\infty) = \phi_\infty$.

This does *not* say that there exist convergent sequences, or that there is *at most one* such subsequence (and thus the sequence converges).

The assumptions we have made do imply that the sequence is *asymptotically regular*, i.e.

$$\|x^{(k+1)} - x^{(k)}\| \to 0.$$

This implies the set of accumulation points, if nonempty, is either a single point or a continuum (a connected and closed set). And all accumulation points have the same function value.

Zangwill's Theorem can be extended to point-to-set maps $A : \Omega \rightarrow 2^{\Omega}$ and sequences of the form

$$x^{(k+1)} \in A(x^{(k)}),$$

but going in that direction will lead us too far astray.

Also, for computation purposes, we need point-to-point maps anyway (and point-to-set maps generally have continuous selections).

Theorem (Ostrowski). Suppose $A : \Omega \rightarrow \Omega$ is differentiable and the sequence $x^{(k+1)} = A(x^{(k)})$ converges to, say, x_∞. If $\lambda(x_\infty)$, the modulus of the largest eigenvalue of $\mathcal{D}A(x_\infty)$, is less than one, then the sequence converges linearly with rate $\lambda(x_\infty)$.

Thus

$$\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x_\infty\|}{\|x^{(k)} - x_\infty\|} = \lambda(x_\infty) < 1.$$

This has as a special possibility that $\lambda(x_\infty) = 0$, in which case we have *super-linear* and, under some additional regularity conditions, *quadratic* convergence.

If $\lambda(x_\infty) = 1$ we have *sub-linear* convergence, often intolerably slow.

Block Relaxation

It is often helpful to partition the variables over which we are minimizing into two or more blocks of variables.

$$\min_{x_1 \in \Omega_1} \cdots \min_{x_s \in \Omega_s} \phi(x_1, \cdots, x_s)$$

Block relaxation algorithms cycle through the blocks, minimizing over one block, while keeping the others fixed at their current values.

$$
\begin{aligned}
x_1^{(k+1)} &= \mathop{\mathbf{argmin}}_{x_1 \in \Omega_1} \phi(x_1, x_2^{(k)}, \cdots, x_{s-1}^{(k)}, x_s^{(k)}), \\
x_2^{(k+1)} &= \mathop{\mathbf{argmin}}_{x_2 \in \Omega_2} \phi(x_1^{(k+1)}, x_2, \cdots, x_{s-1}^{(k)}, x_s^{(k)}), \\
&\vdots \\
x_s^{(k+1)} &= \mathop{\mathbf{argmin}}_{x_s \in \Omega_s} \phi(x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_{s-1}^{(k+1)}, x_s).
\end{aligned}
$$

Of course block relaxation is only interesting if the subproblems are easier to solve than the original problem. This can happen because of the structure of the constraint sets, but more commonly because of the functional form of the specification we are fitting.

Coordinate Relaxation is a special case, in which each block only contains a single variable. It is useful in linear and quadratic programming, and in solving large sparse linear systems and large sparse eigenvalue problems. And in *Iterative Proportional Fitting*.

Zangwill and Ostrowski usually apply. The case of two blocks without constraints

$$\min_{x \in \mathbb{R}^n} \min_{y \in \mathbb{R}^m} \phi(x, y)$$

is especially interesting. Think, for example,

$$\min_{\theta \in \mathbb{R}^n} \min_{\beta \in \mathbb{R}^m} \log |\Sigma(\theta)| + (y - X\beta)' \Sigma^{-1}(\theta)(y - X\beta).$$

See Oberhofer and Kmenta, Econometrika, 1974.

In block relaxation the linear convergence rate is given by the largest eigenvalue of

$$\mathcal{M} = \mathcal{D}_{yy}^{-1} \mathcal{D}_{yx} \mathcal{D}_{xx}^{-1} \mathcal{D}_{xy}$$

where

$$\begin{bmatrix} \mathcal{D}_{xx} & \mathcal{D}_{xy} \\ \mathcal{D}_{yx} & \mathcal{D}_{yy} \end{bmatrix}$$

is the Hessian of the loss function at the solution.

Block-relaxation has as a special case *Alternating Least Squares*, in which a least squares loss function is minimized over two or more blocks of variables. *Factor Analysis* and *Non-metric Multidimensional Scaling* provide examples.

$$\min_{X\in\mathbb{R}^{n\times p}}\min_{\Delta\geq 0}\mathbf{tr}\,(R-XX'-\Delta)'(R-XX'-\Delta),$$

$$\min_{X\in\mathbb{R}^{n\times p}}\min_{\Delta\in\mathcal{K}\cap\mathcal{S}}\mathbf{tr}\,(\Delta-\text{dist}(X))'(\Delta-\text{dist}(X)).$$

Augmentation

Suppose the loss function we try to minimize has a representation of the form

$$\phi(x)=\min_{y\in Y}\psi(x,y).$$

We then minimize loss ϕ by applying block relaxation to the *augmented* loss function ψ.

Finding a suitable augmentation and then using block relaxation defines an *augmentation algorithm* (which generalizes the EM algorithm).

Augmentation algorithms are natural in the case of missing data (unbalanced ANOVA, factor analysis, SVD with missing cells). The missing data are introduced as additional variables. In a least squares context we then use simply

$$\sum_{i\in\mathcal{I}}(y_i-f(x_i,\theta))^2=\min_{z_i=y_i\text{ for }i\in\mathcal{I}}\sum_{i\in\mathcal{I}\cup\mathcal{J}}(z_i-f(x_i,\theta))^2.$$

For augmentation algorithms the linear convergence rate can be written as the largest eigenvalue of the ratio of the Hessian of the loss function and the partial Hessian of its augmentation.

$$\mathcal{M}=I-\{\mathcal{D}_{xx}\psi\}^{-1}\mathcal{D}_{xx}\phi.$$

This uses the obvious fact that

$$\mathcal{D}\phi(x)=\mathcal{D}_x\psi(x,y(x)).$$

The study of convergence rates are important, because they are the basis of acceleration techniques and because block relaxation techniques without modifications can be very slow.

This is also the reason why there has been a lot of research on accelerating the EM algorithm.

Part II: Majorization Algorithms

Majorization Algorithms

For *majorization algorithms* (De Leeuw, from 1977) we construct special type of augmentations.

We say that $\psi : \Omega \to \mathbb{R}$ *majorizes* $\phi : \Omega \to \mathbb{R}$ at $y \in \Omega$ if

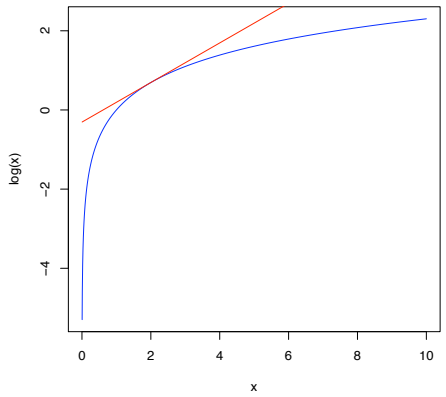
- $\phi(x) \leq \psi(x) \qquad \forall x \in \Omega,$
- $\phi(y) = \psi(y).$

The point y is called the *support point* of the majorization.

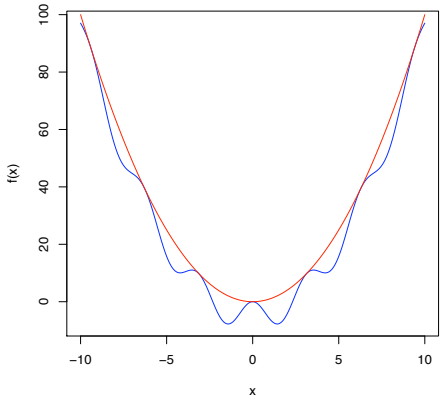
Thus the majorization function at y is always above the function that it majorizes, but it touches that function in y, which is why we call y the *support point* of the majorization. There can be many support points.

A majorization with a single support point is a *strict majorization at* y. In that case we have $\phi(x) < \psi(x)$ for all $x \neq y$.

Below we see *log(x)* strictly majorized by the (tangent) line *log(y) +(x-y)/y* at $y = 2$.



The function $\phi(x) = x^2 - 10\sin^2(x)$ is majorized by the quadratic x^2 and it has support points at all multiples of π.



Majorizing Differentiable Functions

If ψ majorizes ϕ at y then clearly $\psi - \phi$ has its minimum in y. Thus if both functions are differentiable we have

$$\mathcal{D}\psi(y) = \mathcal{D}\phi(y).$$

And if both functions are twice-differentiable also

$$\mathcal{D}^2\psi(y) \geq \mathcal{D}^2\phi(y).$$

We say that a function $\psi : \Omega \otimes \Omega \to \mathbb{R}$ is a *majorization scheme* for $\phi : \Omega \to \mathbb{R}$ if $\psi(\bullet, y)$ majorizes ϕ for each $y \in \Omega$.

For a majorization scheme

$$\phi(x) \leq \psi(x, y) \qquad \forall x, y \in \Omega,$$
$$\phi(x) = \psi(x, x) \qquad \forall x \in \Omega.$$

and thus

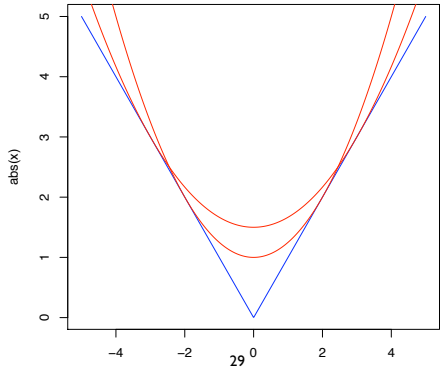
$$\phi(x) = \min_{y \in \Omega} \psi(x, y),$$

which shows majorization provides augmentation.

The function $|x|$ is majorized in $y \neq 0$ by

$$\psi(x|y) = \frac{1}{2|y|}(x^2 + y^2).$$

We draw this for $y=2$ and $y=3$.



Observe we sometimes write $\psi(x|y)$ for $\psi(x, y)$ to emphasize the different roles of x and y.

A majorization algorithm, like any augmentation algorithm, consists of two steps. First we find the majorization (the E step of EM) then we minimize the majorization (the M step of EM).

Because of this Ken Lange (2000) has proposed the name *MM algorithms* (Majorization-Minimization). Willem Heiser (1995) has proposed *Iterative Majorization.*

The key result needed for Zangwill's Theorem is provided by the *sandwich inequality*.

$$\phi(x^{(k+1)}) \leq \psi(x^{(k+1)}|x^{(k)}) \leq \psi(x^{(k)}|x^{(k)}) = \phi(x^{(k)}).$$

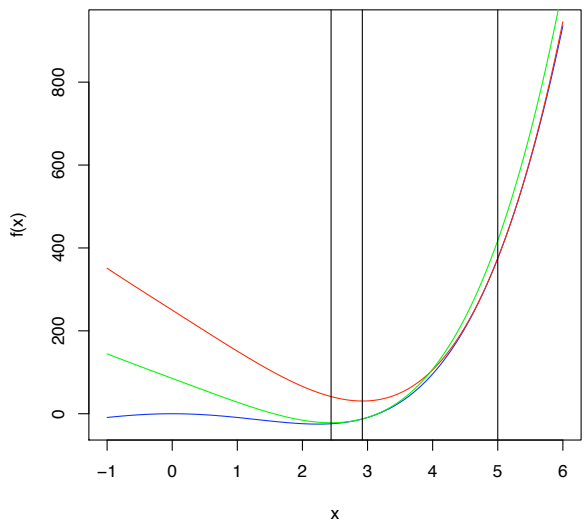
If we have strict majorization, or if we strictly improve the majorizing function, then the inequalities are strict, and we generate a decreasing sequence of loss function values.

By Zangwill, we do not actually have to *minimize* the majorizer. It suffices to have a continuous map A that *decreases* the majorizer -- this will force the sandwhich inequality and thus convergence.

In the EM world this generalization is called GEM, so we could use GMM. The key condition is that $\psi(A(x), x)) < \psi(x)$ for all x.

Let's analyze a simple artificial example. Take $\phi(x) = x^4 - 10x^2$. Because $x^2 \geq y^2 + 2y(x - y) = 2yx - y^2$ we see that $\psi(x, y) = x^4 - 20yx + 10y^2$ is a suitable majorization scheme. The majorization algorithm is $x^{(k+1)} = \sqrt[3]{5x^{(k)}}$.

At $x^{(0)} = 5$ we have the red majorization $\psi(x|5)$. It is minimized at $x^{(1)} \approx 2.92$ with $\psi(x^{(1)}|5) \approx 30.70$ and $\psi(x^{(1)}) \approx -12.56$. Then $\psi(x|x^{(1)})$ is green and has a minimum at $x^{(2)} \approx 2.44$, where $\psi(x^{(2)}|x^{(1)}) \approx -21.79$ and $\phi(x^{(2)}) \approx -24.1$. We are rapidly getting close to the local minimum at $\sqrt{5}$, where ϕ is -25. The convergence rate at this point is $\frac{1}{3}$.



Tricks of the Trade

First trick: Majorization schemes can often be derived from elementary inequalities.

We illustrate this with an example from multidimensional scaling (De Leeuw, 1977). The problem is to minimize *stress*

$$\sigma(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(\delta_{ij} - d_{ij}(X))^2$$

over all *n x p configurations*.

Here $d_{ij}(X)$ is the Euclidean distance between rows i and j of the configuration matrix X. Thus

$$d_{ij}^2(X) = (e_i - e_j)'XX'(e_i - e_j) = \mathbf{tr}\ X'A_{ij}X,$$

where $A_{ij} = (e_i - e_j)(e_i - e_j)'$.

Both the *weights* w_{ij} and the *dissimilarities* δ_{ij} are supposed to be known numbers.

Define

$$V = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} A_{ij}.$$

Then

$$\sigma(X) = 1 + \mathbf{tr}\, X'VX - 2 \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \delta_{ij} d_{ij}(X).$$

By *Cauchy-Schwarz*

$$d_{ij}(X) \geq \frac{1}{d_{ij}(Y)} \mathbf{tr}\, X' A_{ij} Y.$$

In particular if $w_{ij}\delta_{ij} \geq 0$ for all i and j we have

$$\sigma(X) \leq 1 + \mathbf{tr}\, X'VX - 2\mathbf{tr}\, X'(Y)Y,$$

where

$$B(Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \frac{\delta_{ij}}{d_{ij}(Y)} A_{ij}.$$

This provides us with a majorization scheme and with the algorithm

$$X^{(k+1)} = V^{-1} B(X^{(k)}) X^{(k)}.$$

Here is another example. The EM algorithm is designed for functions of the form

$$\phi(x) = -\log \int f(x,z)dz.$$

Using *Jensen's Inequality*

$$\phi(x) - \phi(y) = -\log \frac{\int f(x,z)dz}{\int f(y,z)dz} =$$

$$= -\log \frac{\int f(x,z) \frac{f(x,z)}{f(y,z)} dz}{\int f(y,z)dz} \leq -\frac{\int f(y,z) \log \frac{f(x,z)}{f(y,z)} dz}{\int f(y,z)dz}.$$

Letting

$$f(z|y) = \frac{f(y,z)}{\int f(y,z)dz}$$

we derive the majorization scheme

$$\psi(x|y) = \phi(y) - \int f(z|y) \log f(x,z)dz +$$

$$+ \int f(z|y) \log f(y,z)dz$$

Thus in step *k +1* of the algorithm we minimize the majorization function by minimizing

$$-\int f(z|x^{(k)}) \log f(x,z)dz.$$

Second trick: Majorizations schemes can be derived from convexity considerations.

If ϕ is concave then

$$\psi(x|y) = \phi(y) + z'(x - y)$$

with $z \in \partial\phi(y)$ any subgradient of ϕ at y is a (linear) majorization scheme.

This says that concave functions are majorized by their tangents.

The next two results are due to Lange and De Pierro. They apply the definition of convexity to obtain *separable* majorizations.

Suppose $\gamma : \mathbb{R}^1 \to \mathbb{R}^1$ is convex, $w \geq 0$, and Ω is the positive orthant. Define $\phi(x) = \gamma(w'x)$. Then

$$\psi(x|y) = \frac{1}{w'y} \sum_{i=1}^{n} w_i y_i \gamma(w'y \frac{x_i}{y_i})$$

defines a majorization scheme for ϕ.

Suppose $\gamma : \mathbb{R}^1 \to \mathbb{R}^1$ is convex. Define $\phi(x) = \gamma(w'x)$. Then any choice of v in the unit simplex

$$\psi(x|y) = \sum_{i=1}^{n} v_i \gamma\{\frac{w_i}{v_i}(x_i - y_i) + w'y\}$$

defines a majorization scheme for ϕ.

Third trick: Majorization schemes can be derived from Taylor's Theorem.

If ϕ is twice-diffentiable and there is a matrix H such that $\mathcal{D}^2\phi(x) \leq H$, then

$$\psi(x|y) = \phi(y) + (x-y)'\mathcal{D}\phi(y) + \frac{1}{2}(x-y)'H(x-y)$$

defines a majorization scheme.

This is known as Uniform Quadratic Majorization or UQM. Of course not all functions have bounded second derivatives.

For UQM the algorithmic map is defined by

$$x^{(k+1)} = \underset{x \in \Omega}{\textbf{argmin}} \, (x - z(x^{(k)}))' H (x - z(x^{(k)})),$$

where the *target* $z^{(k)}$ is defined by

$$z(x^{(k)}) = x^{(k)} - H^{-1} \mathcal{D}\phi(x^{(k)}).$$

Thus algorithms based on UQM lead to solving a sequence of weighted least squares problems.

Often the easiest (although not necessarily the best) way to bound the second derivatives is to use a scalar bound

$$\mathcal{D}^2 \phi(x) \leq kI,$$

where it is sufficient to choose k equal to any upper bound for the largest eigenvalue of the Hessian.

UQM is the most common way Taylor's Theorem is used in majorization. There are some (rare) cases where bounding the cubic term makes practical sense.

Best Quadratic Majorization

Uniform quadratic majorization (UQM), where the bound on the Hessian does not depend on y, can often be improved.

The trick is to explicitly take y into account and to compute the best quadratic approximation at this location.

A quadratic

$$\psi(x) = \phi(y) + (x - y)' \mathcal{D}\phi(y) + \\ + \frac{1}{2}(x - y)' A (x - y)$$

majorizes ϕ at y if and only if

$$\frac{1}{2}(x - y)' A (x - y) \geq \\ \geq \phi(x) - \phi(y) - (x - y)' \mathcal{D}\phi(y)$$

for all x.

This defines an infinite set of linear inequalities that A must satisfy, and thus a convex set that A must be in. For *best quadratic majorization (BQM)* we want the smallest A in the convex set.

There are various ways to define "small" for a matrix, but the most common definition uses the Loewner ordering, where $A \geq B$ if $A - B$ is positive semi-definite.

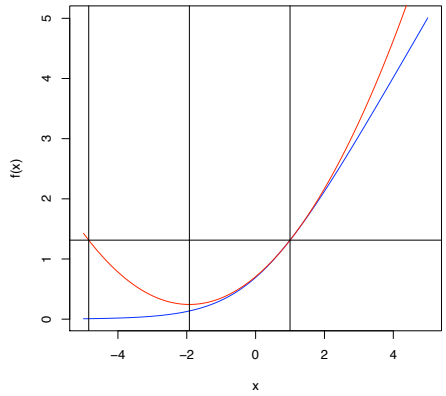
Multivariate BQM has not really been explored yet, and offers an interesting research area.

Finding the BQM is easier if we restrict A to be scalar, and BQM's for the one-dimensional case have been investigated recently for various functions that are important in statistics (Van Ruitenbeek, Groenen, De Leeuw, Lange).

The important thing to realize is that in general the BQM bound will depend on y, and it will always be as least as good as a uniform bound for the second derivative. So BQM will generally improve UQM.

Over-relaxation

Both UQM and BQM can be *over-relaxed*. Consider the example below.



The function (in blue) is majorized at $y = 1$ by the quadratic in red. The quadratic has its minimum at $x = -1.92$ and the usual majorization step would be to take that as the next iteration.

But we decrease the majorization function by taking any update of the form $\lambda\hat{x} + (1 - \lambda)y$, where $\hat{x}$ is the minimizer of the quadratic and $0 < \lambda < 2$.

The *over-relaxed update*, proposed by De Leeuw and Heiser (1980), takes $\lambda = 2$ and update $2\hat{x} - y$.

This changes the convergence rate from κ to $|2\kappa - 1|$. If κ is close to one, then $|2\kappa - 1| \approx \kappa^2$, and convergence is twice as fast (at no cost).

Over-relaxation works in all cases where $\kappa > \frac{1}{3}$. This is usually the case. More sophisticated accelerations that aim to make

$$|\lambda\kappa + (1 - \lambda)| = 0$$

are sometimes also worth studying. We have some recent research that will make the majorization algorithm for MDS about ten times faster.

Intermezzo: Speeding up MDS

Suppose the iterations $x^{(k+1)} = A(x^{(k)})$ converge to x_∞. Suppose in addition that the eigenvalues κ_i of $\mathcal{D}A(x_\infty)$ are between zero and one, with the smallest one κ_n equal to zero and the largest one κ_1 strictly less than one. Suppose moreover that A is *self-scaling*, in the sense that $A(\theta x) = A(x)$ for all x and all real θ.

Then, by Ostrowski, the convergence rate is κ_1.

The convergence rate of the relaxed iterate

$$\lambda A(x^{(k)}) + (1 - \lambda)x^{(k)}$$

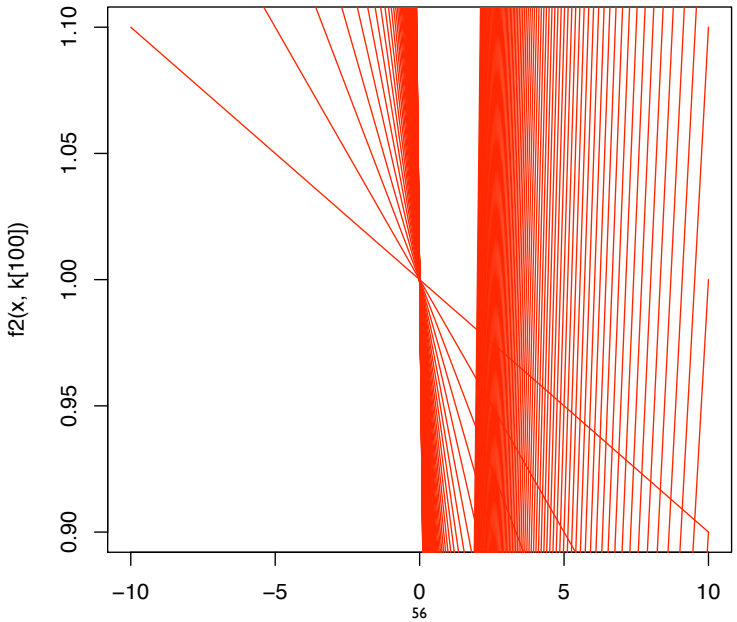
is

$$\nu(\lambda) = \max_{i=1}^{n} |\lambda\kappa_i + (1 - \lambda)|.$$

And the optimum rate in the relaxed family is

$$\min_\lambda \nu(\lambda).$$

In MDS this is attained at $\lambda = \frac{2}{2 - \kappa_1}$ where it is equal to $\frac{\kappa_1}{2 - \kappa_1}$.



If $\kappa_1 = 1 - \epsilon$, with ϵ small, then the optimal λ is approximately $2 - 2\epsilon$ and the optimal rate is approximately $(1 - \epsilon)^2 = \kappa_1^2$.

Unfortunately the relaxed update, while monotone, is not self-scaling. To fix this we now look at

$$\lambda A(A(x^{(k)})) + (1 - \lambda)A(x^{(k)})$$

which is self-scaling. And therefore gives the same iterates as the family

$$A(A(x^{(k)})) + \mu A(x^{(k)})$$

Now the optimal rate is

$$\min_{\mu} \max_{i=1}^{n} |\kappa_i^2 + \mu\kappa_i|.$$

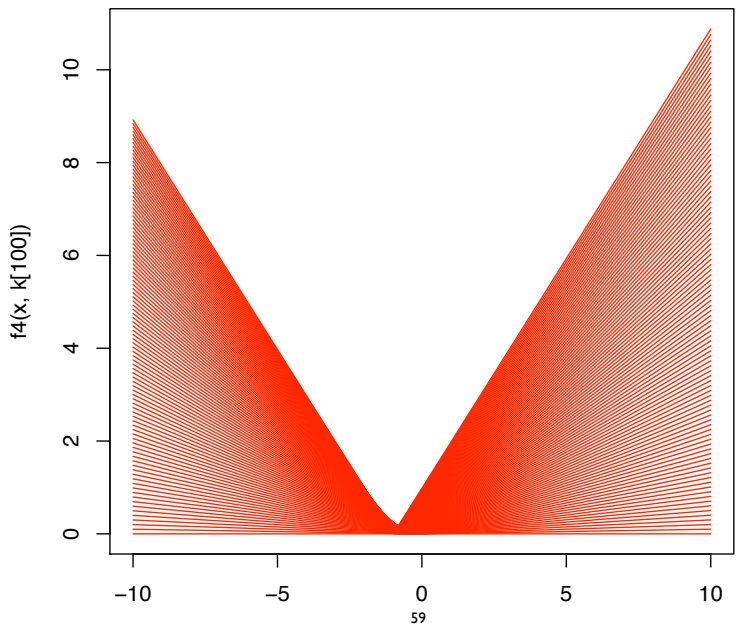
This is attained for

$$\mu = -\max_{i=1}^{n} \frac{\kappa_1^2 + \kappa_i^2}{\kappa_1 + \kappa_i},$$

and (for this i) it is equal to

$$\kappa_1\kappa_i \frac{\kappa_1 - \kappa_i}{\kappa_1 + \kappa_i}.$$

We do not really want to compute this.



But we can use $\mu = -\kappa_1$, for which we find the rate

$$\max_{i=1}^{n} |\kappa_i^2 - \kappa_i\kappa_1| = \kappa_1^2 \max_{i=1}^{n} \frac{\kappa_i}{\kappa_1}(1 - \frac{\kappa_i}{\kappa_1}) \leq \frac{1}{4}\kappa_1^2.$$

This is an enormous speed-up, especially for slow convergence.

With rate .99 we need 230 iterations for an additional decimal of precision, with rate 0.25 only 1.67 iterations.

The algorithm is still pretty simple. In one iteration we start with $x^{(k)}$ and compute $y^{(k)}=A(x^{(k)})$ and $z^{(k)}=A(y^{(k)})$. We then compute

$$\lambda^{(k)} = \frac{\|z^{(k)} - y^{(k)}\|}{\|y^{(k)} - x^{(k)}\|}$$

and

$$x^{(k+1)} = z^{(k)} - \lambda^{(k)} y^{(k)}.$$

dim	diss	eps	non	rel	two
10	250	1e-6	515	118	31
10	250	1e-10	879	300	43
3	500	1e-10	455	261	9
3	500	1e-10	774	432	16
3	500	1e-16	1697	885	23
15	45	1e-6	115	63	15
15	45	1e-10	214	114	22
15	45	1e-10	236	137	22
60	1000	1e-6	390	445	64

Part III: Logits, Probits, Tobits

In this section we apply the basic majorization results to likelihood functions containing logits, probits and tobits.

We limit ourselves, for the moment, to regression with a binary outcome -- simple from the algorithmic point of view but extremely important from the practical point of view.

Regression with ordered or unordered multicategory outcomes is next.

Define

$$f(x) = \log(1 + \exp(x)).$$

Then

$$f'(x) = \frac{\exp(x)}{1 + \exp(x)} = \Psi(x),$$

where

$$\Psi(x) = \frac{1}{1 + \exp(-x)},$$

is the logistic cdf.

Also

$$f''(x) = \Psi(x)(1 - \Psi(x)),$$

from which we see that

$$0 < f''(x) \leq \frac{1}{4},$$

and thus

$$f(x) \leq f(y) + \Psi(y)(x - y) + \frac{1}{8}(x - y)^2.$$

But we can do better using BQM theory.

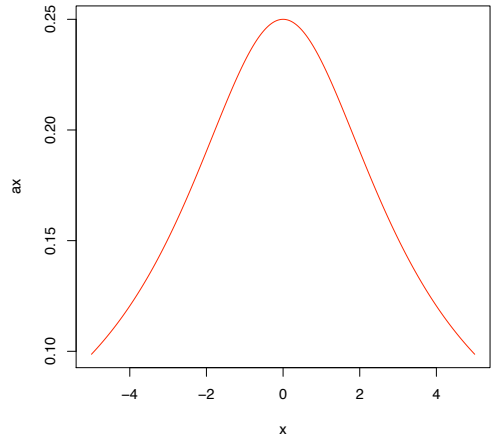
Theorem (Jaakola-Jordan).

$$f(x) \leq f(y) + \Psi(y)(x - y) + \frac{1}{2}A(y)(x - y)^2,$$

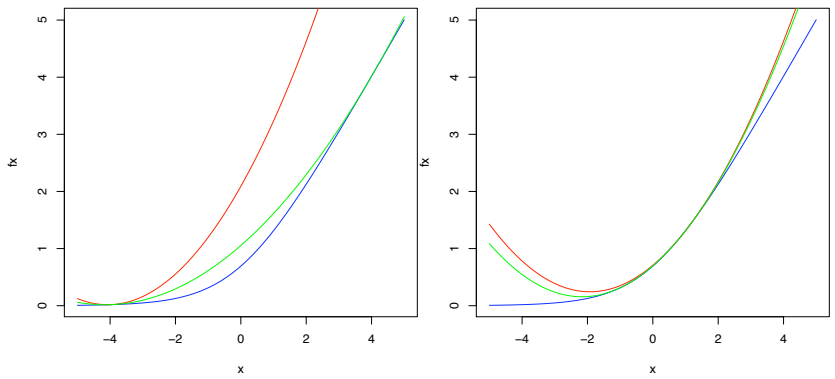
where

$$A(y) = \frac{2\Psi(y) - 1}{2y}.$$

The BQM bound in the logistic case is plotted below. There is no improvement for $y = 0$, but a great deal of improvement in the tails.



UQM (red) and BQM (green) in the logistic case.



Observe that the BQM majorizations have *two* support points: one at y and one at $-y$.

Now apply this to logistic regression. The negative log-likelihood with regressors x_i and binary responses y_i is

$$\Delta(\beta) = \sum_{i=1}^{n} \log(1 + \exp(\tilde{x}_i'\beta)),$$

where

$$\tilde{x}_i = \begin{cases} x_i & \text{if } y_i = 0, \\ -x_i & \text{if } y_i = 1. \end{cases}$$

Quadratic majorizers are of the form

$$\psi(\beta|\tilde{\beta}) = \Delta(\tilde{\beta}) + g(\tilde{\beta})'X(\beta - \tilde{\beta}) + \\ + \frac{1}{2}(\beta - \tilde{\beta})'X'V(\tilde{\beta})X(\beta - \tilde{\beta}),$$

where $g_i(\tilde{\beta}) = \Psi(\tilde{x}_i'\tilde{\beta})$ and $V(\tilde{\beta})$ is diagonal with elements either equal to 1/4 for UQM or equal to $A(\tilde{x}_i'\tilde{\beta})$ for BQM.

Alternative we can also use a scalar UQM bound equal to the largest eigenvalue of $X'X/4$. This is usually, of course, a rather poor approximation.

The majorization algorithm is

$$\beta^{(k+1)} = \beta^{(k)} - (X'V(\beta^{(k)})X)^{-1}X'g(\beta^{(k)}),$$

and the convergence rate is the largest eigenvalue of

$$\mathcal{M} = I - (X'V(\beta^{(k)})X)^{-1}X'WX,$$

where W is $\Psi(\tilde{x}_i'\beta)(1 - \Psi(\tilde{x}_i'\beta))$ evaluated at the solution.

UQM means less work per iteration, BQM means fewer iterations.

Example: Maxwell's percentage of inveterate liars in five age groups.

bound	# iterations	rate
BQM	<10	.0810
UQM	<10	.1192
UQM - scalar	>2000	.9917

Example: Lee's Cancer Remission data, 6 predictors, 27 patients. Un-safeguarded Newton does not converge from most starting points. We start with all regression coefficients equal to one.

bound	# iterations	rate
BQM	275	.9600
UQM	1475	.9929
UQM - scalar	>100,000	1.0000
BQM-overrelaxed	115	.9200
UQM-overrelaxed	731	.9858

Probit Regression

For binary probit regression we use the basic result that for $f(x) = -\log \Phi(x)$, with Φ the normal cdf, we have $0 \leq f''(x) \leq 1$. This provides the constant for UQM, and it turns out that in this case UQM does provide the BQM as well.

For the negative log-likelihood we find

$$\Delta(\beta) = -\sum_{i=1}^{n} \log \Phi(\tilde{x}_i'\beta).$$

The majorization algorithm is

$$\beta^{(k+1)} = \beta^{(k)} - (X'X)^{-1}X'g(\beta^{(k)}),$$

where now g is the *Mills Ratio*

$$g_i(\beta) = -\frac{\phi(\tilde{x}_i'\beta)}{\Phi(\tilde{x}_i'\beta)}.$$

As we said, no BQM acceleration is available for probit regression.

Tobit Regression

The tobit negative log-likelihood in the simplest case is of the form

$$\Delta(\beta) = \frac{1}{2}\sum_{i\in\mathcal{J}_1}(y_i - x_i'\beta)^2 - \sum_{i\in\mathcal{J}_2}\Phi(x_i'\beta).$$

Clearly the same majorizations as for the probit case can be applied to the second term, and a simple unweighted iterative least squares algorithm is the result.

Multicategory (Polytomous) Probits

Probit regression models for polytomous (ordered) data have a negative log-likelihood of the form

$$\Delta(\alpha,\beta) =$$

$$= -\sum_{i=1}^{m}\sum_{j=1}^{m} y_{ij}\log[\Phi(\alpha_j + x_i'\beta) - \Phi(\alpha_{j-1} + x_i'\beta)],$$

where $-\infty = \alpha_0 < \alpha_1 < \cdots < \alpha_{m-1} < \alpha_m = +\infty$ are the *thresholds* and $Y=\{y_{ij}\}$ is the *indicator*.

Majorization theory is dictated here by two results.

Theorem. Suppose $-\infty \leq \alpha < \beta \leq +\infty$ and

$$f(x) = -\log[\Phi(\beta + x) - \Phi(\alpha + x)]$$

Then $0 < f''(x) < 1$.

Theorem. Suppose x is fixed and define

$$h(\alpha,\beta) = -\log[\Phi(\beta + x) - \Phi(\alpha + x)]$$

Then for all $\alpha < \beta$ we have $0 < \mathcal{D}^2 h(\alpha,\beta) < \infty$.

This says that if we are minimizing over the regression coefficients, then we can use UQM (which is also BQM), as in the binary probit model.

But unfortunately there is no upper bound for the second derivatives of the loss function as a function of the thresholds. If thresholds get close to each other, the second derivatives go to infinity. Fortunately the loss function *is* convex in the thresholds.

This strongly suggests the use of block relaxation with two blocks of parameters.

We alternate improving the regression coefficients for fixed thresholds using quadratic majorization and improving the thresholds for fixed regression coefficients using some safeguarded version of Newton's method.

So far, this is easier said than done.

Multicategory (Polytomous) Logits

There are various ways to define logit regression models for polytomous data.

For ordered data we can use

$$\Delta(\alpha,\beta) = \\ = -\sum_{i=1}^m\sum_{j=1}^m y_{ij}\log[\Psi(\alpha_j+x_i'\beta)-\Psi(\alpha_{j-1}+x_i'\beta)].$$

In IRT this is known as the *Graded Response Model*.

This model has not been analyzed yet in terms of majorization theory. It will be interesting to study UQM and BQM, and to establish how second derivatives with respect to the thresholds behave.

For *unordered* polytomous data we can define logistic regression by

$$\Delta(\beta) = -\sum_{i=1}^m\sum_{j=1}^m y_{ij}\log\frac{\exp(x_{ij}'\beta)}{\sum_{\ell=1}^m\exp(x_{i\ell}'\beta)}.$$

UQM is based on the following result.

Theorem. Suppose

$$\pi_j(x) = \frac{\exp(x_j)}{\sum_{\ell=1}^m\exp(x_\ell)},$$

$$f(x) = -\sum_{j=1}^m y_j\log\pi_j(x),$$

then

$$0 \leq \mathcal{D}^2 f(x) = \Pi(x) - \pi(x)\pi(x)' \leq \frac{1}{2}I$$

Some Geometry

Interpreting our optimization problems as maximum likelihood estimation methods is correct, but possibly misleading. It is generally more natural to think of the algorithm as finding approximate solutions to large inconsistent systems of linear inequalities.

If we can find β such that $x_i'\beta > 0$ for all $y_i = 1$ then we can make loss equal to zero by using $\lambda\beta$ with $\lambda \to \infty$. Both for binary logit and probit.

For ordered polytomous logits and probits we want the probability of the k^{th} interval to be the largest whenever $y_{ik} = 1$. This has no obvious interpretation in terms of linear inequalities, but for *unordered* logits we want $(x_{ik} - x_{ij})'\beta > 0$ for all j if $y_{ik} = 1$.

Thus perfect solutions are defined by systems of inequalities. They correspond with zero loss and with parameter estimates wandering off to infinity. The inconsistency of the systems for real data keep the solutions away from infinity.

Part IV: Logit and Probit Component and Factor Analysis

Distance Association Models

Suppose $Y=\{y_{ij}\}$ is an n x m table of observed frequencies. We are interested in models of the form

$$\mathbf{E}(y_{ij}) = \alpha_i\beta_j \exp(\eta(x_i, y_j)),$$

where α and β are the *row and column effects* (a.k.a. the *main effects*) and where X and Y are configurations of points in $\mathbb{R}^p$. The *combination rule* η specifies, in some form, the *similarity* between row object i and column object j.

We now look at a class of models that generalize simple and multiple correspondence analysis, but also the RC model and quasi-symmetry models for cross tables, the Rasch model for item analysis, and various forms of logit and probit component and factor analysis.

The various versions of these models can all be handled by UQM methods, reducing them to sequences of least squares problems of various types.

Different combination rules lead to different forms of the association model. So far we have studied

$$\eta(x_i, y_j) = x_i' y_j,$$
$$\eta(x_i, y_j) = -\|x_i - y_j\|^2,$$
$$\eta(x_i, y_j) = -\|x_i - y_j\|.$$

These are the *inner product*, *negative squared distance*, and *negative distance* rules.

It should be emphasized that

$$\exp(-\|x_i - y_j\|^2) =$$
$$= \exp(-\|x_i\|^2)\exp(-\|y_j\|^2)\exp(2x_i' y_j),$$

which means the inner product rule with row and column effects is equivalent to the negative squared distance rule with row and column effects.

This equivalence is no longer true, however, if we decide not to include one or both of the main effects in the model.

The models also have a *quasi-symmetric version* in which $X = Y$ (of course this is only relevant for square tables) and a *symmetric version* in which in addition also $\alpha = \beta$.

The symmetric and quasi-symmetric versions are often used for input-output, import-export, stimulus recognition, and confusion data in which underlying symmetry is masked by bias or size parameters.

We now need a rule to measure the distance between an observed and an expected table. We use the *Negative Poisson Log-Likelihood* (or Deviance) for this purpose.

$$\Delta(\Lambda) = \sum_{i=1}^{n}\sum_{j=1}^{m}(\lambda_{ij} - y_{ij}\log\lambda_{ij}),$$

where

$$\lambda_{ij} = \alpha_i\beta_j\exp(\eta(x_i, y_j)).$$

As an aside: we do *not* say that the data come are realizations of independent Poisson variables and that we compute maximum likelihood estimates of the parameters.

We just use the Poisson likelihood to measure distances between tables.

But in most cases I am familiar with the Poisson assumption does not make much sense. If it does, then, yes, we are computing maximum likelihood estimates.

First, in minimizing the deviance we use block relaxation combined with majorization. We can find optimal values for the marginal effects α and β by the usual iterative proportional fitting methods. So the only thing we need majorization for is to improve $\eta_{ij}(X, Y) = \eta(x_i, y_j)$.

There are three blocks: improve row-effects, improve column-effects, and improve the distance interactions. The blocks are alternated in the usual way.

In fact we discuss two different majorizations.

Consider the situation where $\eta_{ij} = \eta(x_i, y_j)$ is non-positive, as it is for the distance and squared distance rules. Then

$$\exp(\eta_{ij}) \leq \exp(\tilde\eta_{ij}) + \exp(\tilde\eta_{ij})(\eta_{ij} - \tilde\eta_{ij}) + \frac{1}{2}(\eta_{ij} - \tilde\eta_{ij})^2$$

because the second derivative is the exponent of a non-positive number and is thus never larger than one.

Now substitute this majorization, complete the square, and collect terms. Then in a majorization step we have to minimize

$$\sigma(X,Y)=\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\beta_j(\eta(x_i,y_j)-z_{ij})^2,$$

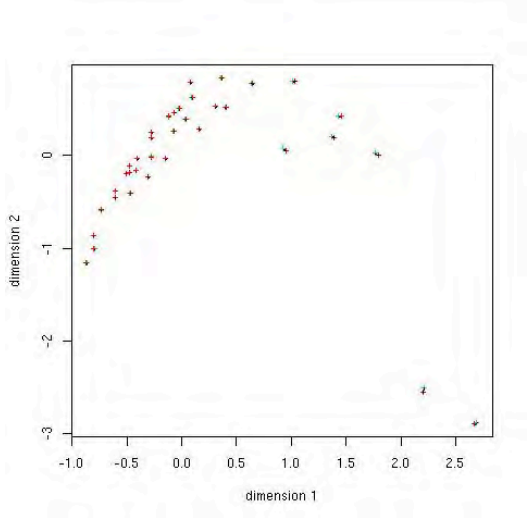
where the target z_{ij} is defined by

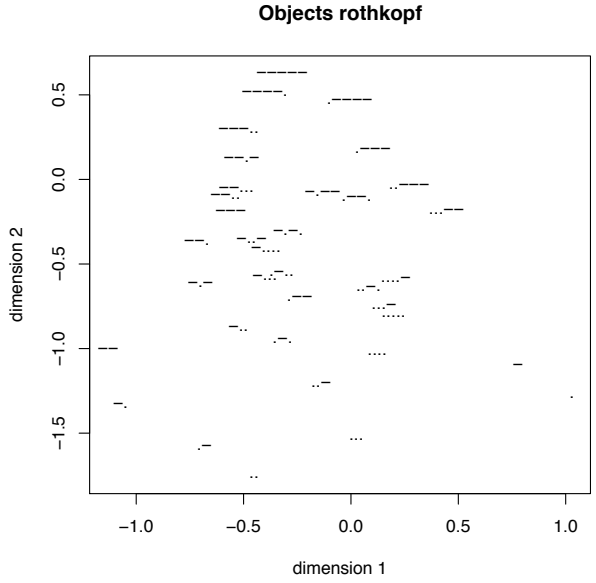
$$z_{ij}=\tilde{\eta}_{ij}+\frac{y_{ij}-\alpha_i\beta_j\exp(\tilde{\eta}_{ij})}{\alpha_i\beta_j}.$$

But we know how to minimize (or decrease) this least squares loss function. Both for the distance rule and for the squared distance rule we can use any number of existing iterative multidimensional scaling algorithms (some of which are again based on majorization).

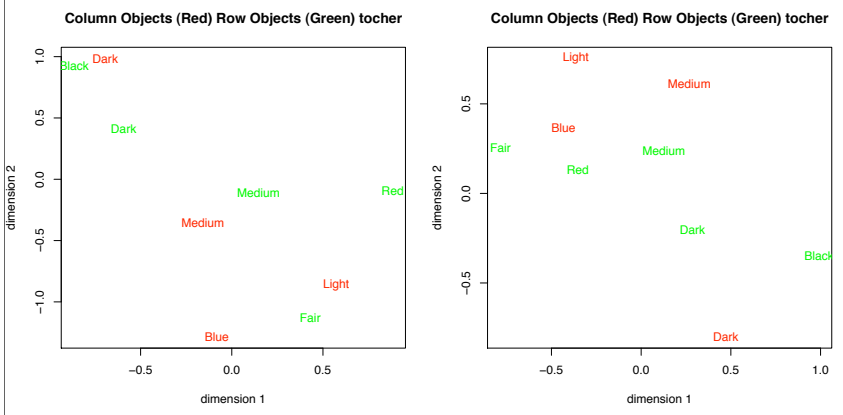
Observe, however, that there is no guarantee that the target is non-positive. This means we need a multidimensional scaling algorithm that can cope with negative dissimilarities.

Rothkopf Morse Code Data





Tocher Eye and Hair Color



dist dist2

This algorithm cannot deal directly with the inner product rule, because it depends on non-positivity.

We could fit the innner product model by fitting squared distances rules with bias, but there is a more direct way (which also makes it possible to fit inner product rules without bias).

The idea is to minimize $\min_{\alpha} \Delta(\alpha, \beta, X, Y)$. The minimum is attained at

$$\alpha_i = \frac{y_{i\star}}{\sum_{j=1}^m \beta_j \exp(\eta_{ij})}$$

and it is equal (except for some irrelevant constants) to

$$\sum_{i=1}^n \sum_{j=1}^m y_{ij} \frac{\beta_j \exp(\eta_{ij})}{\sum_{\ell=1}^m \beta_\ell \exp(\eta_{i\ell})}.$$

But this is of the form we used in the case of unordered polytomous logit models. Using the basic majorization result for that case we find

$$\sigma(X, Y) = \sum_{i=1}^n y_{i\star} \sum_{j=1}^m (\eta(x_i, y_j) - z_{ij})^2,$$

where

$$z_{ij} = \eta_{ij}(\tilde{X}, \tilde{Y}) + 2(p_{j|i} - \pi_{j|i}(\tilde{X}, \tilde{Y}, \beta)),$$

and $p_{j|i}$ and $\pi_{j|i}$ are the row-normalized y_{ij} and λ_{ij}.

Voronoi Models for Categorical PCA

The idea of removing the row effects is more generally applicable. Consider the situation in which we have m categorical variables, variable j has k_j categories, and the data are coded as *indicator matrices*). Consider the Poisson deviance

$$\Delta(\alpha,\beta,X,Y)=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell=1}^{k_j}\{\lambda_{ij\ell}-y_{ij\ell}\log\lambda_{ij\ell}\},$$

with

$$\lambda_{ij\ell}=\alpha_{i\ell}\beta_{j\ell}\exp(\eta(x_i,y_{j\ell})).$$

Minimizing out the $\alpha_{i\ell}$ gives the loss function

$$-\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell=1}^{k_j}y_{ij\ell}\log\frac{\beta_{j\ell}\exp(\eta(x_i,y_{j\ell}))}{\sum_{\nu=1}^{k_j}\beta_{j\nu}\exp(\eta(x_i,y_{j\nu}))}$$

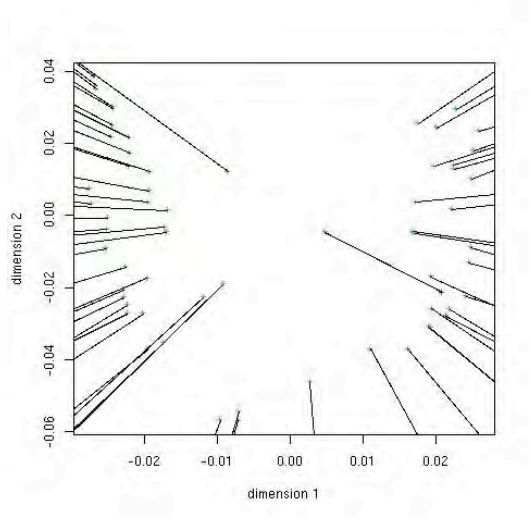
which can be majorized by our logistic methods.

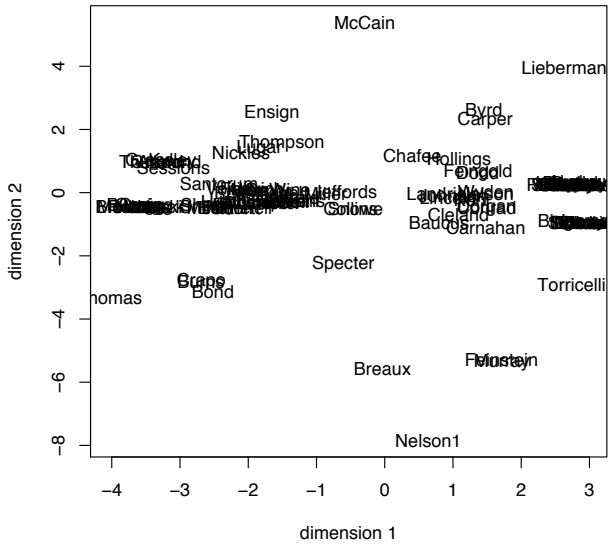
This gives a basis for a far-reaching generalization of the non-linear multivariate analysis methods of Gifi (1990). We can handle the same data and restrictions, but choose combination rules.

The geometry of these solutions can be discussed in the same way as we did in regression. We want $\beta_{j\ell}\exp(\eta(x_i,y_{j\ell}))$ to be the largest where $g_{ij\ell}=1$.

For no bias and the squared distance or distance rule thus means we want x_i to be closest to the y_{jl} of the category it is in, or we want x_i to be in the correct Voronoi cell. For the inner product model without bias the cells are cones with apex at the origin and each x_i should be in the correct cone. For binary data the Voronoi cells are half spaces, which separate the "aye's" from the "nay's".

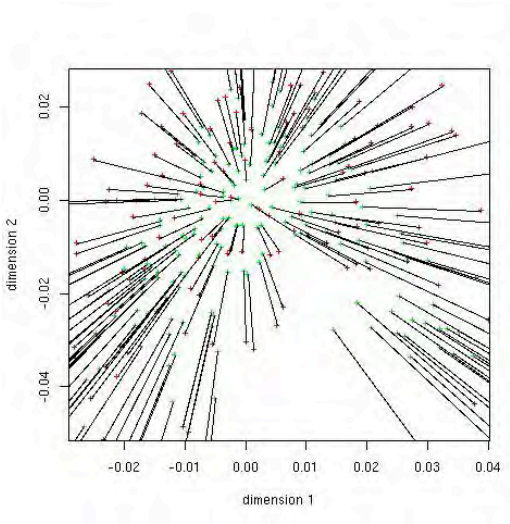
Senate Data



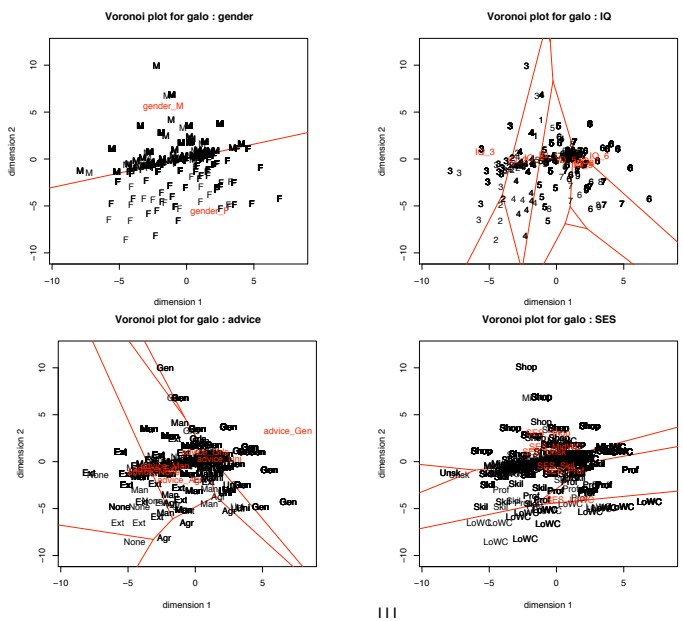


109

GALO Data

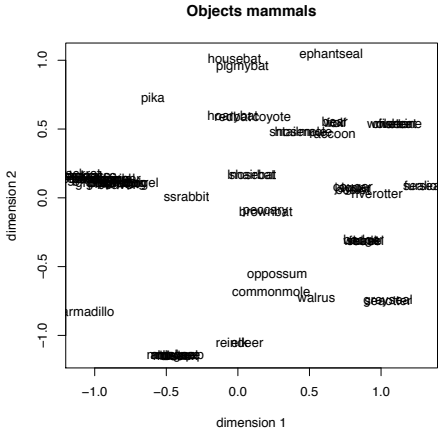


110

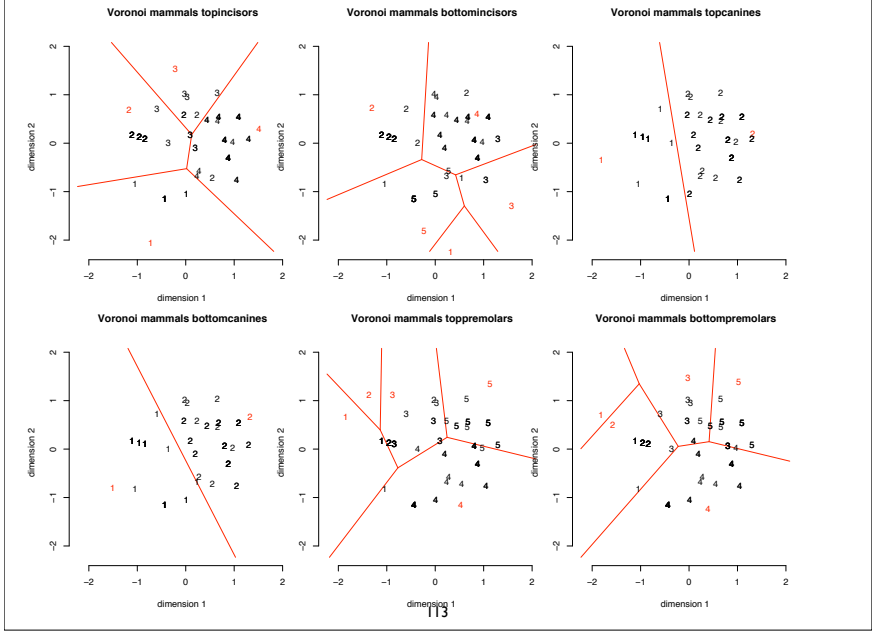


111

Dentition Data



112



Logit Models for Single-Peaked Data

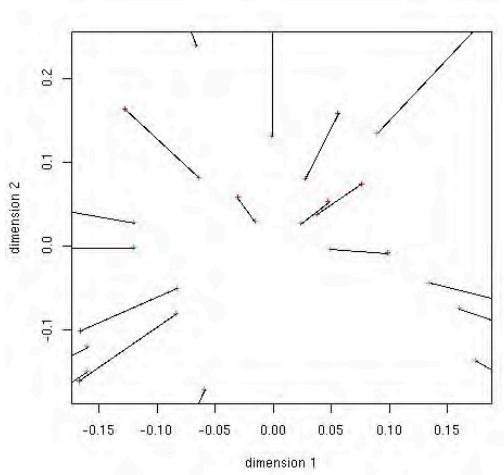
$$\mathcal{D}(X,Y,\xi) = -2\left\{\sum_{(i,j)\in I_1} \log\frac{\beta_{ij}(\xi)\exp(\phi(x_i,y_j))}{1+\beta_{ij}(\xi)\exp(\phi(x_i,y_j))} + \right.$$

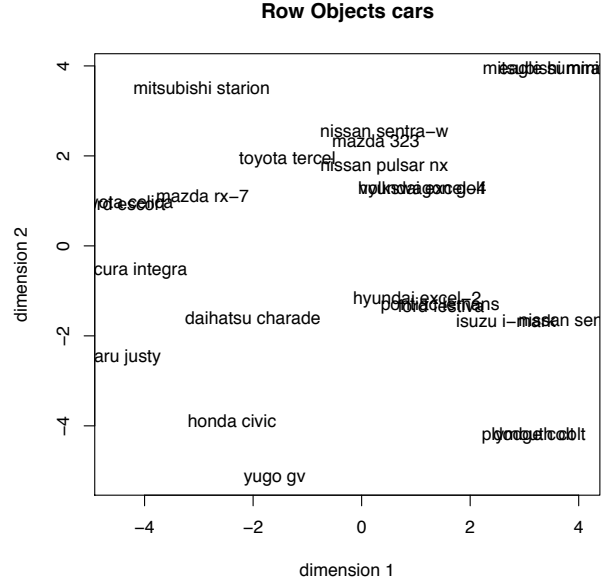
$$\left. + \sum_{(i,j)\in I_0} \log\frac{1}{1+\beta_{ij}(\xi)\exp(\phi(x_i,y_j))}\right\}$$

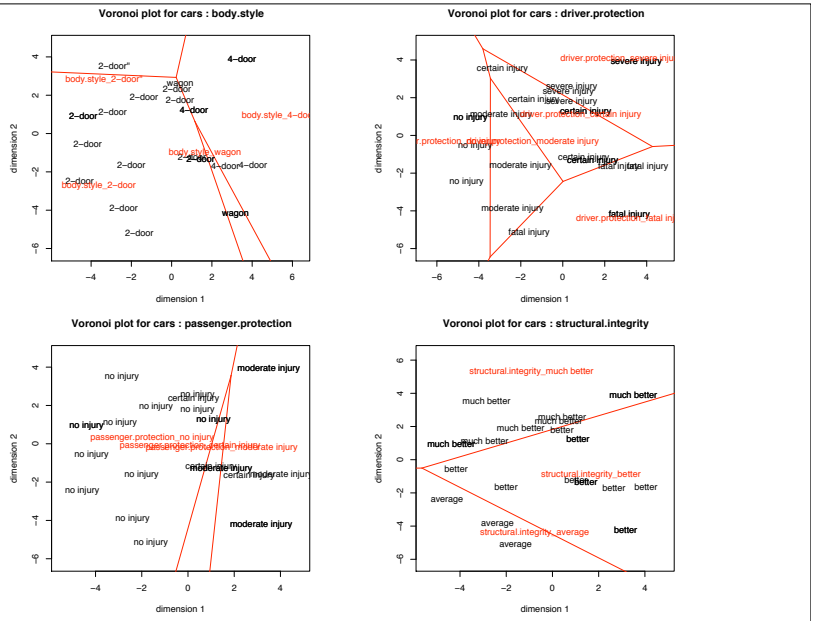
We also specify $\beta_{ij}(\xi) = \exp(\gamma_{ij}(\xi))$, with

$$\gamma_{ij}(\xi) = \sum_{s=1}^{p} z_{ijs}\xi_s.$$

Cars Data







The geometry in this case is somewhat different from our previous geometry, where using distances leads to Voronoi cells.

In this cases using distances leads to inequalities which says that the "aye's" are in a circle (sphere) around y_j, while the "nay's" are outside the circle (sphere).

Of course the regression specification of the bias parameters also applies to our previous models, where we just did not explore this.

Multivariate Probit Models

$$\mathcal{D}(\tau, \theta) =$$

$$= -2 \sum_{i=1}^{n} f_i \sum_{j=1}^{m} \sum_{\ell=1}^{k_j} y_{ij\ell} \log \left[\Phi(\tau_{j,\ell} + \eta_{ij}(\theta)) - \Phi(\tau_{j,\ell-1} + \eta_{ij}(\theta)) \right].$$

This does not introduce any new theory, but it has not been implemented in sufficient detail to produce examples.

Part V: Marginal Maximum Likelihood

The method so far uses parameters x_i for objects.

Thus, in the classical statistical sense, they have *incidental parameters*, and this may lead to bias in the estimation of the structural parameters y_j.

A common way out of this dilemma is to use *random score models* and *marginal maximum likelihood estimation*. Unfortunately this introduces multidimensional integrals into the loss function, and causes all kinds of Bayesian mischief.

Instead of

$$-\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell=1}^{k_j} y_{ij\ell}\log\frac{\beta_{j\ell}\exp(\eta(x_i,y_{j\ell}))}{\sum_{\nu=1}^{k_j}\beta_{j\nu}\exp(\eta(x_i,y_{j\nu}))}$$

we use

$$-\sum_{j=1}^{m}\sum_{\ell=1}^{k_j} y_{\bullet j\ell}\log\int\frac{\beta_{j\ell}\exp(\eta(x,y_{j\ell}))}{\sum_{\nu=1}^{k_j}\beta_{j\nu}\exp(\eta(x,y_{j\nu}))}\pi(x)dx.$$

There are several ways to handle these loss functions.

First: estimate the density non-parametrically. This means it becomes a step-function, and we can use EM (i.e. majorization) in a block-relaxation process.

Second: approximate the known density by quadrature. This reduces to our previous algorithms, except now the X are known quadrature points.

Third: apply MCMC to compute maximum posterior probability estimates. This is very much

, but I'll leave it to others to explore this line of research.

Fourth: Use (quadratic majorization to find) variational approximations to the integrand and then integrate (assuming, for instance, a normal density).