PSEUDO-VORONOI DIAGRAMS FOR MULTICATEGORY EXPONENTIAL REPRESENTATIONS

JAN DE LEEUW

ABSTRACT. Generalizations of the planar Voronoi diagram for exponential distance and inner product models with or without bias parameters are discussed.

1. INTRODUCTION

In several recent pre-publications, for example De Leeuw [2004, 2006b], we have generalized the logistic PCA approach of De Leeuw [2006a] to multicategory data. The basic data are *m indicator matrices* Y_j with n rows and k_j columns, as in Gifi [1990]. Thus Y_j is binary, and its rows add up to one (unless there are missing data, in which case they add up to zero). Indicator Y_j corresponds with a variable that takes on k_j different values.

The *n objects* and the $\sum_{j=1}^m k_j$ *levels* of all variables are represented as points in $\mathbb{R}^p$, where more often than not $p = 2$. Coordinates for the objects are in the $n \times p$ matrix A, while those for the levels of variable j are in the $k_j \times p$ matrix B_j.

We choose a *combination rule* ϕ on $\mathbb{R}^p \otimes \mathbb{R}^p$, such that $\phi(a_i, b_{j\ell})$ quantifies how similar or close object i and category ℓ of variable j are. The most common combination rules are the inner product $\phi(a_i, b_{j\ell}) = a_i'b_{j\ell}$, the negative distance $\phi(a_i, b_{j\ell}) = -\|a_i - b_{j\ell}\|$, and the negative squared distance $\phi(a_i, b_{j\ell}) = -\|a_i - b_{j\ell}\|^2$.

Date: April 26, 2006.

2000 *Mathematics Subject Classification.* 62H25.

Key words and phrases. Multivariate Analysis, Correspondence Analysis.

The general idea is to minimize the deviance

$$\mathcal{D}(A,B)=-2\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{\ell=1}^{k_j}y_{ij\ell}\log\frac{\beta_{j\ell}\exp(\phi(a_i,b_{j\ell}))}{\sum_{\nu=1}^{k_j}\beta_{j\nu}\exp(\phi(a_i,b_{j\ell}))} \tag{1}$$

over the *configurations* A and B, and possibly over the *bias vectors* β_j. The bias vectors are the response probabilities of the levels if $\phi(a_i,b_{j\ell})=0$. They are chosen to be positive, and within each variable they add up to one. They correspond with the marginal effect of the levels in a log-linear model.

The deviance has a likelihood interpretation, but that interpretation is rather far-fetched. A geometrical interpretation is much more natural [De Leeuw, in press]. Consider the inequalities

$$y_{ij\ell}(\beta_{j\ell}\exp(\phi(a_i,b_{j\ell}))-\beta_{j\nu}\exp(\phi(a_i,b_{j\nu}))>0. \tag{2}$$

Or, equivalently, if $y_{ijl}=1$ then $\beta_{j\ell}\exp(\phi(a_i,b_{j\ell}))$ must be the largest of the $\beta_{j\nu}\exp(\phi(a_i,b_{j\nu}))$ for all ν. In the case in which there is no bias and we use the negative distance, or the negative squared distance, this says that each object i must be closest to the category of the variable that this object i scores in. Or, if we make a Voronoi plot [Okabe et al., 2000] of the k_j levels of variable j, and plot the n objects on top, then each object must be in the correct Voronoi cell. If a variable is binary ($k_j=2$, then the Voronoi cell is a half-space, and there is an hyperplane separating the objects in the first category from those in the second category. This special case explains the popularity of the binary version of the technique in political science [Clinton et al., 2004] and item response theory [Reckase, 1997].

In general, of course, we cannot expect to find A and B that satisfy the inequalities (2) precisely. That is why we need a badness-of-fit measure such as the deviance to minimize. What we can show, however, is that if (2) is solvable, then $\inf_{A,B}\mathcal{D}(A,B)=0$. Thus we can get arbitrarily close to perfect fit, basically by letting some or all points diverge to infinity in the appropriate direction.

2. Problem

The problem we study in this note is to draw a plot, for each variable, that shows in how far inequalities (2) are satisfied by a solution we have computed. In the unbiased case with negative distances or squared distances, the problem is basically solved. Drawing a Voronoi diagram is a much studied problem, and can be done efficiently for a large number of points. There is code in various R packages such as `deldir` that interfaces to efficient C libraries. But our problem is more general, because we want to incorporate the effect of bias, and we also want to be able to deal with the inner product combination rule.

We shall produce a simple *R* program that draws these generalized Voronoi plots. Since in our applications n will usually be very small, and p will usually be 2, we do not really care about optimizing the order of the computations, we just want computations that can be done conveniently in R.

Both for the inner product model and for the squared distance model the inequalities (2) are linear. For the distance model, without the square, this unfortunately is not the case, and we do not discuss that combination rule here.

So the general structure of the problem we are trying to solve is as follows. Suppose $f_i(x) = u_i'x - v_i$ are n linear functions on $\mathbb{R}^2$. We want to compute and draw the n polygons

$$\mathcal{M}_i = \{x | f_i(x) = \max_{k=1}^{n} f_k(x)\},$$

which of course partition the plane. Without any real loss of generality we assume that $u_i \neq u_j$ if $i \neq j$.

3. Algorithm

The basic idea of the algorithm is to enumerate the line segments in the diagram, and then to draw them. Only very elementary calculations are needed.

First define the lines

$$\mathcal{L}_{ij} = \{x | f_i(x) = f_j(x)\} = \{x | (u_i - u_j)'x = v_i - v_j\}.$$

Let

$$e_{ij} = (v_i - v_j)\frac{u_i - u_j}{(u_i - u_j)'(u_i - u_j)}$$

so that $(u_i - u_j)'e_{ij} = v_i - v_j$, and let

$$f_{ij} = \begin{bmatrix} u_{j2} - u_{i2} \\ u_{i2} - u_{j2} \end{bmatrix}$$

so that $(u_i - u_j)'f_{ij} = 0$. The line $\mathcal{L}_{ij}$ consists of all points $e_{ij} + \lambda f_{ij}$, with $-\infty < \lambda < +\infty$.

All segments in the diagram will be on these lines. Now look for the interval $[\lambda_{min}, \lambda_{max}]$ such that for all $k \neq i, j$ we have

$$u_i'(e_{ij} + \lambda f_{ij}) - v_i \geq u_k'(e_{ij} + \lambda f_{ij}) - v_k$$

Remember that for all λ we have $u_i'(e_{ij} + \lambda f_{ij}) - v_i = u_j'(e_{ij} + \lambda f_{ij}) - v_j$.

Collecting terms gives

$$\lambda(u_i - u_k)'f_{ij} \geq (v_i - v_k) - (u_i - u_k)'e_{ij},$$

and thus

$$\max_{k \in K_+} \frac{(v_i - v_k) - (u_i - u_k)'e_{ij}}{(u_i - u_k)'f_{ij}} \leq \lambda \leq \min_{k \in K_-} \frac{(v_i - v_k) - (u_i - u_k)'e_{ij}}{(u_i - u_k)'f_{ij}}.$$

Here K_+ is the set of all $k \neq i, j$ for which $(u_i - u_k)'f_{ij} > 0$ and K_- is the set of all $k \neq i, j$ for which $(u_i - u_k)'f_{ij} < 0$. If $(u_i - u_k)'f_{ij} = 0$ and $(v_i - v_k) - (u_i - u_k)'e_{ij} > 0$, then the interval is empty and we go to the next k. Because testing for zero is always tricky in floating point computations, this is where we may have some numerical problems. If the interval is not empty, we store it and/or draw the segment. After looping through all $i < j$ and all $k \neq i, j$ we are done.

4. Example

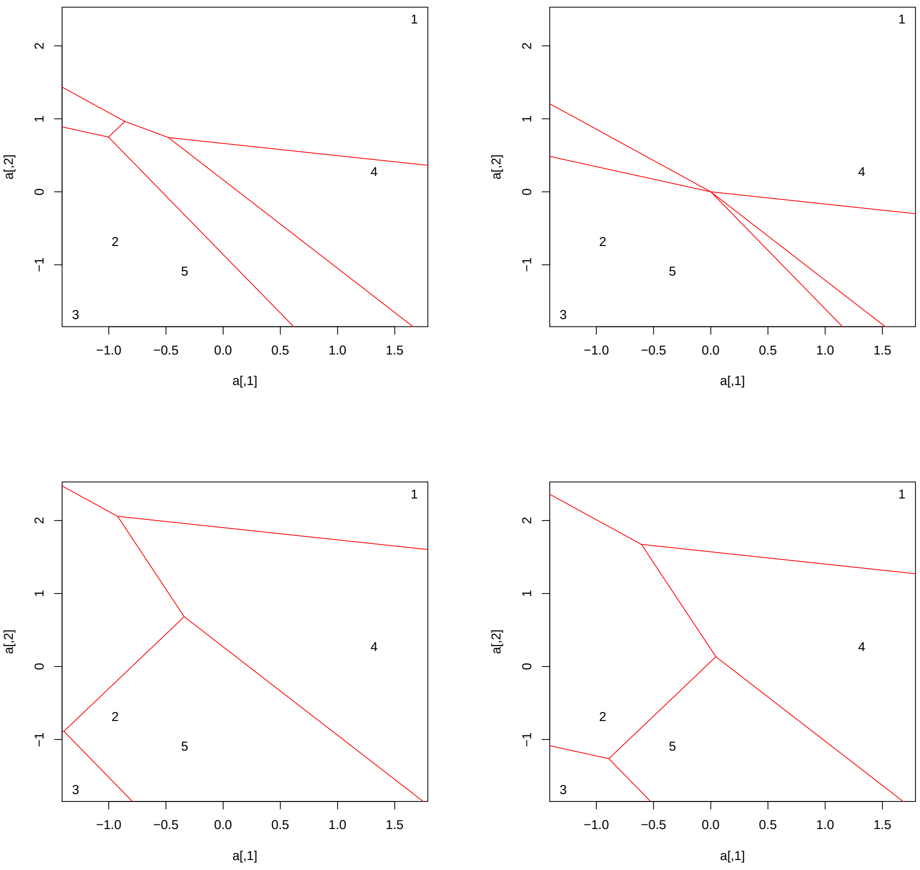
We use a small artificial example. U has 5 points, and there is a bias vector b. The bias gets transformed for the inner product rule to the inequality right-hand side by $v_j = -\log(b_j)$, and for the squared distance rule to $v_j = \frac{1}{2}(\sum_{s=1}^{p} u_{js}^2 - \log(b_j))$. This is the only difference between the two rules, as far as the algorithm is concerned.

```
> u
           [,1]       [,2]
[1,]  1.6708897  2.3665436
[2,] -0.9439634 -0.6832961
[3,] -1.2893324 -1.6860028
[4,]  1.3200185  0.2748327
[5,] -0.3349960 -1.0912174
> b
[1] 0.06666667 0.13333333 0.20000000 0.26666667 0.33333333
```

We then call the function four times, with and without bias, and inner products or squared distances. The calling code is

```
pdf("ipbi.pdf")
drawEdges(u,b)
dev.off()
pdf("ipni.pdf")
drawEdges(u,b=rep(1,5))
dev.off()
pdf("sdbi.pdf")
drawEdges(u,b,fit="sd")
dev.off()
pdf("sdni.pdf")
drawEdges(u,b=rep(1,5),fit="sd")
dev.off()
```

The four plots show the diagrams. If there is no bias (on the right), then the inner product rule (at the top) partitions the space into cones with apex at the origin. And the squared distance rule (at the bottom) leads to the usual Voronoi diagram, in which the lines are all segments of perpendicular bisectors.



Appendix A. Code

```
makeEdges<-function(a,b,fit="ip",verbose=FALSE) {
if (fit=="ip") c<--log(b) else c<-(-log(b)+rowSums(a^2))/2
n<-length(b); lns<-matrix(0,0,8)
for (i in 1:(n-1)) {
    for(j in (i+1):n) {
        dd<-a[i,]-a[j,]; dc<-c[i]-c[j]; ss<-sum(dd^2)
        if (is.null(ss)) next()
        ee<-dc*dd/ss; ff<-c(-dd[2],dd[1])
        xlw<--Inf; xup<-Inf
        for (k in (1:n)[-c(i,j)]) {
            dd<-a[i,]-a[k,]; dc<-c[i]-c[k]
            mum<-sum(dd*ff); mom<-dc-sum(dd*ee)
            if (is.nul(mum) & (mom > 0)) {
                xlw<-Inf; xup<--Inf
                }
            if (mum>0) xlw<-max(xlw,mom/mum)
            if (mum<0) xup<-min(xup,mom/mum)
            if (verbose) {
                cat(formatC(i,digits=3,width=3),
                    formatC(j,digits=3,width=3),
                    formatC(k,digits=3,width=3),
                    "mum ",formatC(mum,digits=4,width=8,format="f"),
                    "mom ",formatC(mom,digits=4,width=8,format="f"),
                    "xlw ",formatC(xlw,digits=4,width=8,format="f"),
                    "xup ",formatC(xup,digits=4,width=8,format="f"),
                    "\n")
                    }
            }
        if (xlw<xup) lns<-rbind(lns,c(i,j,ee,ff,xlw,xup))
        }
    }
return(lns)
}

drawEdges<-function(a,b,fit="ip",far=1000,verbose=FALSE) {
lns<-makeEdges(a,b,fit,verbose=verbose)
p<-dim(lns)[1]; n<-dim(a)[1]
plot(a,type="n"); text(a,as.character(1:n))
for (i in 1:p) {
    ee<-lns[i,3:4]; ff<-lns[i,5:6]
    xlw<-lns[i,7]; if (xlw == -Inf) xlw<--far
    plw<-ee+xlw*ff
    xup<-lns[i,8]; if (xup == Inf) xup<-far
    pup<-ee+xup*ff
    lines(rbind(plw,pup),col="RED")
    }
}

is.nul<-function(x) {
return(abs(x)<1e-10)
}
```

References

J. Clinton, S. Jackman, and D. Rivers. The Statistical Analysis of Roll Call Data. *American Political Science Review*, 98:355–370, 2004.

J. De Leeuw. Nonlinear Principal Component Analysis and Related Techniques. In M. Greenacre and J. Blasius, editors, *Multiple Correspondence Analysis and Related Methods*. Chapman and Hall, in press.

J. De Leeuw. Principal Component Analysis of Binary Data by Iterated Singular Value Decomposition. *Computational Statistics and Data Analysis*, 50(1):21–39, 2006a.

J. De Leeuw. Majorization Methods for Distance Association Models. Technical report, UCLA Department of Statistics, 2006b.

J. De Leeuw. Logistic Homogeneity Analysis. 2004.

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

A. Okabe, B. Boots, K. Sugihara, and S.N. Chiu. *Spatial Tessellations*. Wiley, second edition, 2000.

M.D. Reckase. A Linear Logistic Multidimensional Model. In W.J. Van Der Linden and R.K. Hambleton, editors, *Handbook of Item Response Theory*. Springer, 1997.

Department of Statistics, University of California, Los Angeles, CA 90095-1554

E-mail address, Jan de Leeuw: deleeuw@stat.ucla.edu

URL, Jan de Leeuw: http://gifi.stat.ucla.edu