BEYOND HOMOGENEITY ANALYSIS

JAN DE LEEUW

Abstract. In this paper we propose various extensions of homogeneity analysis. These extensions call all be discussed in terms of the geometrical representation of objects and categories. They complete the system of measurement levels used in older forms of homogeneity analysis and nonmetric principal component analysis, mostly by introducing various possibilities to analyze 'continuously' varying classifications.

This paper was originally Research Report RR-84-08 of the Department of Data Theory, University of Leiden, 1984. An modified version was published as De Leeuw and Van Rijckevorsel [1988]. This version has R code added to do the computations and draw the figures. I corrected some typos. Everything in this paper, except for the pseudo-indicators and the continuous process restrictions, is now implemented in the homals package in R [De Leeuw and Mair, 2009].

1. Introduction

In Gifi [1981a] a large number of multivariate analysis methods is organized in a single general framework. The key method in this system is *homogeneity analysis*, also known as *multiple correspondence analysis*. The Gifi system is inspired by ideas from multidimensional scaling, in particular by the central role of Euclidean distance in the representation of complex multivariate data. The basic data we want to represent geometrically are categorizations of n objects by m variables. Although the assumption that the variables are discrete and assume only a finite number of values is not essential, and can even be made without any practical loss of generality, it is true that in the current versions of

Date: Monday 27th April, 2009 — 23h 1min — Typeset in Lucida Bright.

Key words and phrases. Correspondence analysis, principal component analysis, multidimensional scaling.

homogeneity analysis categorical variables with a small number of categories play a central role. Variables with a large number of possible values, or even "continuous" variables, can be incorporated in theory, but the implementations of the techniques more or less expect a small number of categories. If the number of categories is very large, say close to the number of objects that are classified, then homogeneity analysis as currently implemented [Gifi, 1981b] does not work very well. It will tend to produce unsatisfactory and highly unstable solutions, in which "chance capitalization" is a major source of variation.

There have been various attempts in Gifi [1981a] to make the solutions more stable by imposing restrictions that reflect, in some sense, the prior information we have about the variables. In De Leeuw [1984b] these restrictions are classified into *rank-restrictions*, *cone-restrictions*, and *additivity-restrictions*. Imposing restrictions decreases the number of free parameters. This means, roughly, that there are more data values per parameter, which can consequently be determined in a more stable manner. Rank-restrictions and cone-restrictions make it more easy to deal with variables having a large number of categories, but in several respects their treatment remains somewhat unsatisfactory. In many multidimensional scaling programs there are options for transformation of the variables that are "smooth" or otherwise "continuous". There is no such possibility in the current homogeneity analysis programs. In this paper we shall try to extend the basic geometry of homogeneity analysis in such a way that continuous variables fit in more easily. A fundamental role in this extension is played by the "B-spline basis", which is introduced here in a purely geometrical way (compare also Van Rijckevorsel [1982]). We also introduce, as a further generalization, a "fuzzy" type of B-spline basis, which indicates more clearly how homogeneity analysis generalizes the various forms of nonmetric principal component analysis [De Leeuw, 1982]. Combination of the various options creates a very flexible new type of homogeneity analysis. It is highly unlikely that all possible types will be equally important in practice, in fact we suspect that some of the less restricted forms will again tend to produce highly unstable or even "trivial" solutions. Nevertheless it is satisfactory from a theoretical point of view to show exactly what the choices are that one has to make, even if some of the possible choices may be quite unwise in practical situations.

2. Simple Homogeneity Analysis

We start with a brief recapitalization of the technique of homogeneity analysis, without any of the frills discussed by Gifi [1981a] or De Leeuw [1984a,b]. The data are m variables on n objects, i.e. there are m functions defined on a common domain $\{1, 2, \cdots, n\}$. We suppose that the range of function j has k_j elements, and we code function j by using the $n \times k_j$ *indicator matrix* G_j. Matrix G_j is binary, it has exactly one element equal to one in each row, indicating into which element of the range the object corresponding to this row is mapped, Thus the rows of G_j add up to one, and the matrix $D_j = G_j'G_j$is diagonal, and contains the univariate marginals. If G_j and G_ℓ are indicator matrices of two different jvariables, then $C_{j\ell} = G_j'G_\ell$ is the cross-table of variables j and ℓ, i.e. it contains the bivariate marginals. This notation is illustrated in detail in [De Leeuw, 1973, Chapter 2], Gifi [1981a, Chapter 23], but also in Guttman [1941] and in Burt [1950].

The purpose of homogeneity analysis is to map both objects and variables into low dimensional Euclidean space $\mathbb{R}^p$ (where p is *dimensionality*, chosen by the user). We want to do this in such a way that both objects and categories of the variables are represented as points, and in such a way that an object is relatively close to a category it is in, and relatively far from the categories it is not in. Of course this implies, by the triangle inequality, that objects mostly scoring in the same categories tend to be close, while categories sharing mostly the same objects tend to be close too. The extent to which a particular representation X of the objects and particular representations Y_j of the categories, satisfy the desiderata of homogeneity analysis is measured by a least squares loss function. This is defined as

$$\sigma(X; Y_1, \ldots, Y_m) = \sum_{j=1}^{m} \mathbf{tr}\ (X - G_jY_j)'(X - G_jY_j) \tag{1}$$

In order to prevent certain obvious trivialities we require that the $n \times p$ matrix of *object scores* X is normalized by $u'X = 0$ and $X'X = nI$. Here

u is a vector with all elements equal to one, and I is the identity matrix. We do not normalize the m matrices of *category quantifications* Y_j, which are of order $k_j \times p$. Using (1) and the normalization conventions we can now give a more precise definition of homogeneity analysis. It is to choose a normalized X and $Y_1, \cdots, Y_m$ such a way that (1) is minimized. For additional interpretations of the loss function, in terms of consistency, discrimination, and homogeneity, we refer to Gifi [1981a] and De Leeuw [1984b]. In this paper we more or less ignore the algorithmic and statistical aspects of the homogeneity analysis techniques, and we concentrate on the geometry on which the loss function is based.

3. Pictures of Loss

In Table 1 we have presented a small example with 10 objects and three variables. The objects are 10 cars, the variables are price (in $ 1000), gas consumption (litres per 100 km, on the expresway), and weight (in 100 kg). The data are taken from a larger matrix used by Winsberg and Ramsay [1983, Page 587], who took their data from the April, 1983 issue of *Consumer Report*. In order to prevent possible misunderstandings we must emphasize that Table 1 is not at all representative for data usually analyzed with homogeneity analysis. In fact in most practical applications of the technique the number of objects and the number of variables is much larger. Moreover in our small example all variables are numerical, which is also not typical for most homogeneity analysis applications.

Insert Table 1 about here

The data in Table 1 cannot be used directly in homogeneity analysis. They must first be made discrete or categorical. This is done by grouping the values of the variables into discrete categories, which can, of course, be chosen in many different ways. One possible, fairly crude, categorization is given in Table 2. Observe that there are three cars with *profile* (1,1,1), and two cars with (2,1,2). Thus there are only seven different profiles for these ten cars, out of a possible $3 \times 3 \times 4 = 36$ profiles. A finer discretization would give more possible profiles, more different actual profiles, and also more “empty cells”, i.e. more profiles that do not occur. The finest discretizations is the *ranking* given in Table 3. Here there are $10^3 = 1000$ possible profiles, of which only 10 are in use. Thus 99% of the cells is empty. Observe that in constructing Table 3 from Table 1 we have arbitrarily broken a tie a variable 2 (Chevette and Pontiac Phoenix both score 6.9 in gas consumption).

INSERT TABLE 2 ABOUT HERE

INSERT TABLE 3 ABOUT HERE

Now suppose we choose object scores X in two dimensions, and category quantifications Y_j also in two dimensions. We have plotted the objects scores we have chosen as ten points in Figure 1. Also given in Figure 1 are the three points corresponding with the categories of variable 1, price. To make a picture of loss, for variable 1, we have connected all objects with the category point they belong to, according to variable 1. Loss-component 1 is simply the sum of squares of the line-lengths drawn in figure 1. We can make a similar picture for variable 2, if we also choose Y_2 It is important to realize that we have chosen X and Y_1 completely arbitrary, and not by any optimality considerations. They are not, in any sense, the solutions given by homogeneity analysis. In fact they are merely candidates for the solutions, and it is the purpose of the technique to find better candidates. Another important point is that we can also make "dual" pictures, in which we plot all Y_j as points together with a single object point.

INSERT FIGURE 1 ABOUT HERE

INSERT FIGURE 2 ABOUT HERE

The loss "due to object i" can now be represented by drawing lines from the object point to all category points it is in. Such plots, as well as the plot in Figure 1, are "sub-plots" of a large plot which contains all object-points and all category points, and which has a line for each element equal to one in each indicator matrix. This "super-plot" will generally look somewhat messy, so it is better to present it in "layers". In Figure 2 we have presented the optimal solution computed by homogeneity analysis, i.e. the optimal object scores and the optimal quantifications of the categories of variable 1. It is clear that the line lengths are shorter for the optimal solution. For other types of plots useful in homogeneity analysis we refer to Gifi [1981a,b].

4. Rank Restrictions

In simple homogeneity analysis category quantifications can be anywhere in $p-$space. From Equation (1) it follows that optimal category quantifications are centroids of objects points in the categories. This is illustrated in Figure 2. In fact in Figure 1 category quantifications of variable 1 are also optimal for the given object scores, only the object scores are very far from optimal in this case. Because of the *centroid-property* of optimal category quantifications it follows that their weighted average, with weights equal to the marginal frequencies, is the origin. This is the only restriction on the relative position of the quantifications of the categories within a variable. Now consider the situation in which variables have a range which is ordinal or even numerical. This constitutes a form of prior information which is not used by simple homogeneity analysis, and which consequently may get lost in the representation computed by homogeneity analysis. If we look at Figure 2 the categories of variable 1 are represented in the "correct" order. This is true if we measure order along the horizontal axis, and even more clearly true if we measure order along the "horse-shoe" on which all objects lie. For variable 2, gas consumption, the situation is quite different, however. Only Dodge Diplomat and AMC Eagle are in category 3, which means that the optimal quantification of the category will be the midpoint of the line connecting DD and AE. Category 2 contains CI, OM, and BR, and will be quantified close to CI. Category 1 will be between cluster CC, DC, PH and cluster PP, FM. Thus both on the horse-shoe and on the line the categories will project in the order 1-3-2, which is contrary to our prior information.

Another property of simple homogeneity analysis is that very often two-dimensional plots will occur which are in the form of a horse-shoe. Of course there is nothing which is intrinsically wrong with horse-shoes. It is just that they are somewhat wasteful. They use two dimensions to present an essentially one-dimensional structure. Or, to put it differently, the second best dimension for discriminating the objects turns out to be a nonlinear transformation of the first one. In some cases we may be interested in a more independent second dimension. We shall not analyse the precise reasons for the regular occurrence of the horseshoe in homogeneity analysis. They are given in Gifi [1981a]; De Leeuw [1982]; Schriever [1983]. In this paper we merely discuss geometrically inspired methods which both get rid of the horse-shoe and make it possible to impose our prior information.

Our first basic idea is to do this by imposing *rank-one restrictions*. By this we mean that we require all category quantifications of a variable to be on a line through the origin of $p-$space, with each variable having its own line. In matrix notation this means that we require $Y_j = z_ja_j'$, i.e. the $k_j \times p$ matrix Y_j must be of rank one. In order to distinguish the various types of category quantifications that result from this idea we now call the Y_j *multiple category quantifications*, while the z_j are called *single category quantifications*. The a_j are the *loadings* of variable j. We now minimize the loss function (1), with the provision that for some variables (but not necessarily for all) we use the restrictions $Y_j = z_ja_j'$. Variables for which the restrictions are imposed are called *single variables*, variables without restrictions are *multiple variables*. A program for homogeneity analysis with mixed multiple and single variables is discussed by Gifi [1982].

In order to study the geometry of single variables we expand the corresponding loss component first. This gives

$$\begin{aligned} \mathbf{tr}\,(X - G_jY_j)'(X - G_jY_j) &= \mathbf{tr}\,(X - G_jz_ja_j')'(X - G_jz_ja_j') = \\ &= np - 2a_j'X'G_jz_j + (z_j'D_jz_j)(a_j'a_j'). \end{aligned} \tag{2}$$

Now let $q_j = G_jz_j$, and normalize z_j such that $u'q_j = 0$ and $q_j'q_j = n$. Such normalization is used merely for identification purposes, because z_j only occurs in the product z_ja_j'. Using the normalization we find

$$\mathbf{tr}\,(X - G_jY_j)'(X - G_jY_j) = n(p-1) + (q_j - Xa_j)'(q_j - Xa_j). \tag{3}$$

This shows, in the first place, that single loss cannot possibly be zero if p is larger than one. It is always at least $n(p-1)$. It is equal to $n(p-1)$ if all objects in a category project in the same point on the line through the origin and a_j . Or, to put it differently, if categories define parallel hyperplanes orthogonal through the line defining the variable. All objects in a category must be located in the hyperplane of the category. The elements of z_j are the signed distances to the origin of the category hyperplanes, i.e. the locations of the projections on the line defining the variable. In the case of nonperfect fit the loss is simply the distance of each object point from its category hyperplane, or, more precisely, the squared distance. Figure 3 illustrates this for a particular choice of X, z_1 and a_1 in our small cars example. Again no optimality considerations are used here, in fact we have not even paid attention to the appropriate normalizations. It is clear that rank-one restrictions will tend to make horse-shoes impossible, or at least highly unlikely. It may not be clear yet how they can be used to impose ordinal or numerical prior information. Before we proceed to explaining this, we relate the use of single variables to performing a *principal component analysis*.

Insert Figure 3 about here

Suppose, for the moment, that all variables are single. Collect the q_j in the $n \times m$ matrix Q, with $u'Q = 0$ and $\mathbf{diag}(Q'Q) = nI$. Collect the a_j in the $m \times p$ matrix A. Then (3) implies that

$$\sigma(X; Y_1, \cdots, Y_m) = nm(p-1) + \mathbf{tr}(Q - XA')'(Q - XA'). \tag{4}$$

Now minimizing (4), which is the same thing as minimizing (1) if all variables are single, means performing a *singular value decomposition* on Q, or a principal component analysis on the correlation matrix $R = n^{-1}Q'Q$. Remember, however, that Q depends on the z_j. It is thus not a constant matrix. Minimizing (4) can be interpreted as choosing (single) quantifications of all variables in such a way the Q is as close as possible to a rank p matrix. Or: in such a way that the sum of the p largest eigenvalues of R is as large as possible. Homogeneity analysis, with all variables single, is for this reason also called *nonlinear principal component analysis.* It is nonlinear, not because we use nonlinear approximations, but because optimal nonlinear transformations are computed for all variables. Compare De Leeuw [1982] or Bekker [1982] for a much more extensive discussion.

5. Cone Restrictions

Rank one restrictions induce an order on the categories of the variable, even if we do not know the order beforehand. The induced order is given by the projections on the variable vector, or by the order of the category hyperplanes. In fact the category hyperplanes even introduce a single numerical scale for the categories of a variable, given in the vectorz_j. Now the induced ordinal or numerical information mayor may not correspond with our prior knowledge. We use *cone restrictions* if we impose the constraint that the induced order must be the same as our prior order, and the induced scale must be the same as our prior scale. Numerically these are restrictions on the elements of z_j. Either they must be in the "correct" order, for *single ordinal* variables, or they must be equal to a given normalized vector, for *single numerical* variables. Observe that the type of a variable refers to the constraints we impose, it does not reflect some intrinsic property of the variable. We use the term "cone restrictions" because the feasible choices for z_j form a polyhedral convex cone in k_j-space for ordinal variables, and a one-dimensional subspace, which is sort of degenerate cone, for numerical variables. It is also possible, by the way, to formulate our restrictions in terms of $q_j = G_j z_j$, i.e. in $n-$space. No restrictions on z_j, defining *single nominal* variables, defines a k_j-dimensional subspace in $n-$space. Ordinal and numerical restrictions defines subcones and subspaces of this k_j-dimensional subspace.

If the z_j are completely given, by restrictions taken together with normalizations, then homogeneity analysis becomes identical with principal component analysis. This is, in a sense, one of the endpoints of the continuum of homogeneity analysis techniques. All variables are single numerical. The other endpoint has all variables *multiple nominal*. This is what we have described earlier as simple homogeneity analysis or multiple correspondence analysis. In Figure 4 we give a two dimensional principal component analysis representation of our small example, using the geometry of homogeneity analysis.

Insert Figure 4 about here

Figure 4 results from analyzing Table 2. It is clear, of course, that the analysis of tables 1 and 3 would give different results in general. Table 3 is quite interesting in this respect. For Table 3 the indicator matrices G_j are permutation matrices. If we substitute them in (1) it is obvious that loss can always be made equal to zero by letting X be an arbitrary $n \times p$ matrix, and by setting $Y_j = G_j'X$. Then $G_jY_j = G_jG_j'X = X$. In the same way single nominal variables can always be fitted perfectly. Choose X and a_j arbitrarily, and set $z_j = G_j'Xa_j$Then $q_j = Xa_j$, and loss is minimized by (3). In other words: nontrivial analysis of rankings is possible only if we make all variables either single ordinal or single numerical. It is also interesting to compare the single quantifications in $q_j = G_jz_j$ with the original scores in Table 1. Clearly plotting the elements of q_j versus the original scores will give a step-function. We have discreticized our variables, and as a consequence every object in the same discretization interval gets the same quantification in the q-vector of the variable. The more intervals, the less crude the transformation given by the step-function will be, but no matter how fine we choose the discretization the transformation will always be a step function. This is one of the main reasons why we say that homogeneity analysis as currently implemented by! Gifi [1981a,b] has a discrete bias. Step-functions are perfectly natural for variables which have a small number of possible values to start with, or for purely nominal variables for which we have no prior numerical information. For “continuous ” numerical variables, such as the three variables in our example, transformation by step-functions ignores the prior information that our variable was originally continuous, and can also assume all intermediate values between the end-points. Thus we now now how to incorporate numerical and ordinal information, but we do not know yet how to incorporate “smoothness” into homogeneity analysis. This problem will be discussed below, but first we have to fill a number of gaps that have been left open in the combination of various options we have discussed up to now.

6. Gaps in Gifi

In the previous sections we have discussed single numerical, single ordinal, single nominal, and multiple nominal variables. We did not discuss multiple ordinal and multiple numerical. If only for esthetic reasons it is interesting to investigate if these remaining types of variables can also be given a simple meaning. Moreover we have distinguished single and multiple variables. For single variables we required that $\mathbf{rank}(Y_j)$ was less than or equal to one, For multiple variables there were no rank restrictions, which means that we "required" that $\mathbf{rank}(Y_j)$ was less than or equal to $\min(p, k_j - 1)$. It is $k_j - 1$ and not k_j in this upper bound, because of the fact that the rows of Y_j have a weighted mean of zero. Now if $p = 1$ there is no difference between multiple and single. If $p = 2$ then for variables with more than two categories single requires that $\mathbf{rank}(Y_j)$ is less than or equal to one and multiple that $\mathbf{rank}(Y_j)$ is less than or equal to two. There is no gap between the two options. But for $p = 3$, and k_j larger than three, single requires rank one and multiple requires rank three as the upper bound. Thus there is a gap. We can insert another option, which requires $\mathbf{rank}(Y_j)$ to be less than or equal to two. This general rank restriction, which can be between single and multiple, was already discussed in De Leeuw (1976), but it was not incorporated in the subsequent developments of the Gifi-system.

The loss function, with general rank constraints, can be written as

$$\sigma(X; Y_1, \cdots, Y_m) = \sum_{j=1}^{m} \mathbf{tr}\ (X - G_jZ_jA_j')'(X - G_jZ_jA_j') \tag{5}$$

Here Z_j is $k_j \times r_j$, and Aj is $p \times r_j$ The r_j are the required ranks for variable j. Geometrically the constraint means, of course, that the category quantifications must be in a r_j-dimensional hyperplane through the origin. If $Z_j'D_jZ_j = nI$, then loss for variable j satisfies

$$\sigma_j(X, Y_j) = n(p - r_j) + \mathbf{tr}\ (XA_j - G_jZ_j)'(XA_j - G_jZ_j). \tag{6}$$

If $A = (A_1 \mid \cdots \mid A_m)$ and $Q = (Q_1 \mid \cdots \mid Q_m) = (G_1Z_1 \mid \cdots \mid G_mZ_m)$ then

$$\sigma(X; Y_1, \cdots, Y_m) = nm(p - \overline{r}) + \mathbf{tr}\ (XA - Q)'(XA - Q). \tag{7}$$

This looks very similar to (4), but remember that in (7) each Q_j consists of r_j orthogonal quantifications of the same variable, i.e. of r_j *copies* (compare De Leeuw [1984b]; Tijssen [1985]). Again, geometrically, we have minimum loss if the category points are in an r_j-plane, and all object points are on lines perpendicular to the plane, which cross the plane in the k_j -category points.

General rank restrictions now make it possible to define r_j-nominal, in which there are no further restrictions on Z_j.There is also r_j-numerical, in which the r_j columns of Z_j are known orthogonal k_j-vectors. And, finally, there is r_j-ordinal, in which all columns of Z_j must be in the appropriate order. For r_j-nominal and r_j-numerical we can require, without loss of generality, that $Z_j'D_jZ_j = nI$. For r_j-ordinal such a constraint cannot be imposed, and we have to refrain from normalizing Z_j and/or A_j [1] . It is clear, of course, that general rank constraints, coupled with *measurement restrictions*, generalize our previous notions of single and numerical, and fill the gaps in the system. In fact it opens completely new possibilities: we can require that the first "copy" in Z_j is ordinal, while the remaining copies are nominal, and so on. Again we do not know how practical these new options are. We have discussed them because they fit naturally into the gaps, and also because they can be incorporated without much ado into the homogeneity analysis algorithms that are already there.

7. Pseudo Indicators

A more satisfactory analysis of continuous variables becomes possible if we generalize the notion of an indicator matrix. Suppose we continue to use the same notion of loss, with the same types of restrictions on the category transformations, but we do not soppose that the G_j are indicator matrices. They must still be known $n \times k_j$ matrices, but they need not be binary any more. In a sense we have already gone a step in this direction. If a variable is r_j-numerical, then $Y_j = G_j(Z_jA_j') = (G_jZ_j)A_j'$. Suppose, for instance, that the Z_j are polynomials, orthogonal with respect to the marginals. Then G_jZ_j are orthogonal polynomials in $n-$space, and we can interpret our analysis as an unrestricted analysis using an $n \times r_j$ basis of orthogonal polynomials instead of the indicator

[1] In the current `homals` package [De Leeuw and Mair, 2009] we require for r_j-ordinal variables that only the first column of Z_j is in the correct order, while the other columns are orthogonal to the first. Thus we also have $Z_j'D_jZ_j = nI$ in this case.

matrix G_j. Although this is clearly a valid interpretation, it is not exactly what we have in mind.

In this section we concentrate on so called *fuzzy codings*, collected in *pseudo-indicator matrices*. Fuzzy codings were first introduced, in full generality by Martin [1980a,b], although various special cases were already studied earlier by other investigators. The most important special case is the B-spline basis, which was used by Lafaye de Michaux [1978]; De Leeuw et al. [1981]; Van Rijckevorsel [1982]. Winsberg and Ramsay [1980, 1983] have used I-splines, which are not entirely in the class we discuss, although they are closely related. A pseudo-indicator matrix G_j, to give a precise definition, is an $n \times k_j$ nonnegative matrix whose rows add up to one. Thus the mass, concentrated in a single category for simple indicator matrices, can be spread over the categories for pseudo-indicators. The *bandwidth* of a pseudo-indicator is the largest number of nonzero elements in a row. Thus indicator matrices are characterized as pseudo-indicators with bandwidth unity. Piecewise linear B-splines define pseudo-indicators with bandwidth two, and so on. In this paper we do not care about the origin of the pseudoĕindicators, for this we refer to the publications of Martin and Van Rijckevorsel. We simply assume that data are coded in this way, and we look for the geometrical interpretations of such a coding. In Table 4 we have a fuzzy coding of our small example, which is actually the result of piecewise linear coding. The idea behind our generalization of homogeneity analysis now is that we can combine all our previous options and restrictions with this new coding as well. In particular we can impose rank-constraints, and impose ordinal or numerical restrictions.

Because $p = 2$ in our example it suffices to distinquish single and multiple. Consider multiple nominal. The loss component for variable j vanishes if $X = G_jY_j$. In the coding used in Table 4 each X corresponds with two categories, because the bandwidth in our example is two. The two category quantifications are the endpoints of a line segment, all line segments for a particular variable are connected. The object scores must be on the line segment corresponding to the categories they are in. And not only must they be on the segment, they must also be in a precise location on the segment, where the location is dictated by the masses of the endpoints in the coding. This is indicated in Figure **??**, which is not an optimal solution of any kind, but it is used to illustrate the loss of variable 1 in the coding of Table 4. The points on the two line segments indicate where the cars must be given the coding. and given the location of the endpoints of the segments. In the single case the endpoints must be on the same straight line, and the object points must project on the places fixed by the coding. Thus there are parallel lines perpendicular to the line connecting the category points, which intersect this line at the appropriate places. In the ordinal case the endpoints must be ordered along the line. such that both within-category and between-category quantifications are ordered. In the single nominal case only within-category quantification is ordered (which makes this a somewhat peculiar option, perhaps).

If we study the *transformation* which considers $q_j = G_j z_j$ as a function of the original data values, then transformations from pseudo-indicators will indeed be more smooth than those from indicators. The precise nature of the smoothness depends on the nature of the pseudo-indicators, for instance on the bandwidth. In our example the transformations are continuous and piecewise linear. If we use piecewise quadratic splines, joined in such a way that they are differentiable at the endpoints, then we get more smoothness (and a bandwidth of three). The geometry becomes more complicated, because object points must be at the appropriate places in the triangle spanned by three endpoints. Successive triangles are interlocked, because they have one side in common. And so on, for larger bandwidths, and/or in higher dimensions. It follows from this picture that bandwidth three or more does not combine naturally with single quantification, because single quantification makes the triangles degenerate to straight lines. This is no problem analytically, but it makes the geometry of loss far less interesting. In general we think that for practical purposes a bandwidth larger than two is probably not very interesting, unless data are very well behaved indeed.

8. Process

In the developments so far the data were coded as (pseudo)-indicators, and these pseudo-indicators were fixed during the computations. Now let us look at single ordinal piecewise linear again. We have already seen that the order of the category points on the line is fixed in this case, although their precise location is free. Given the location of the category points, however, the location of the preferred projection of the object points on the line is fixed by the coding. This is what we mean by fixedness of within-category order. This fixedness is contrary to what is called the *primary apporach to ties* in multidimensional scaling literature, and also the *continuous ordinal option* (compare De Leeuw et al. [1976]; Young et al. [1980]; Young [1981]). In this option, which is incorporated in various nonmetric principal component programs, we fix the order between categories but not within categories. Or, geometrically, given the line and the location of the category points on the line, the object-point can project *anywhere* between the end-points of its category. Loss only occurs if they project outside their assigned interval.

Given our previous discussion it is easy to see how the idea of continuous ordinal data can be incorporated easily into our form of homogeneity analysis. The elements of the pseudoěindicators are not considered fixed any more, only the location of the nonzero elements is fixed. Thus we know which elements must be nonzero, we also know that they must be nonnegative and they must add up to one for each row, but their precise values are additional parameters over which the loss function is minimized. In the single ordinal piecewise linear case this gives exactly continuous ordinal data as treated in PRINCIPALS, for instance [Young et al., 1978]. But because we have fitted the possibility of varying the elements of the G_j into our general homogeneity analysis framework, we can combine this option with all other previous options that we already had. It can be combined with multiple quantification, and with single numerical quantification. In this last case it gives the continuous numerical scaling earlier discussed by De Leeuw and Walter [1977]. There is very little need to elaborate on the geometry of the continuous versions. It is basically the same as the discrete geometry, only points are not fixed in intervals, but they can be anywhere in the interval. It becomes perhaps a bit more interesting to use larger bandwidths with single options, because the bandwidth now controls the amount of overlap of the intervals corresponding with the categories. If bandwidth is two, there is no overlap. In bandwidth is three successive categories have one common subinterval, and so on. Multiple options with bandwidth three, in two dimensions, are interpreted in terms of triangles (or convex hulls). Objects in category 1 must be in the convex hull of category points 1, 2, and 3, objects in category 2 in the convex hull of 2,3, and 4, and so on. Successive triangles have one side in common, if they degenerate to line segments this becomes the overlapping subinterval. It is not at all clear (yet) if these conceptually very nice options are useful in practice. Their conceptual nicety may be a bit misleading in this respect. A theorem in Gifi [1981a] is useful to illustrate their limitation. It refers to the continuous ordinal option, with all variables single. The results shows that with this option degenerate solutions, which locate one object very far away from the others, which are collapsed into a single point, will be quite comnon. In fact Gifi shows that in the situation in which objects are a random sample the minimum of loss is almost surely equal to zero if the sample size tends to infinity. We do not know yet how devastating this results is in practice, but it certainly indicates that we have to be careful.

Computationally our new options do not introduce any trouble at all. We must introduce a new subproblem into the alternating least squares cycles of homogeneity analysis in which the G_j are adjusted. This is done for each row of each G_j separately, defining a very small special quadratic programming problem. Of course we have to exert a little self-control in combining our options. We have the possibility, in principle, to take a different bandwidth for each object, or a different rank for each Y_j . In fact, looming large in the distance, is the possibility of further generalizations. We can fix the bandwidth of each variable, for instance, and determine the optimum location of the nonzero elements. This is probably very unwise, because the program output will become almost independent of the data.

It is perhaps convenient to relate existing programs to our general form of homogeneity analysis, in which we choose (a) quantification rank, (b) measurement level, (c) bandwidth, (d) process for each variable separately. HOMALS [Gifi, 1981b] has quantification rank equal to dimensionality, measurement level nominal, bandwith unity, and process discrete. Of course if bandwidth is unity there is no distinction between discrete and continuous process. Ordinary principal component analysis has quantification rank unity, measurement level numerical, bandwidth unity, process discrete. PRINCALS [Gifi, 1982] has quantification rank either one or dimensionality, and measurement level numerical, ordinal, or nominal (but ordinal/numerical cannot occur together with multiple). Bandwidth is unity, and process is discrete. SPLINALS [Van Rijckevorsel, 1982; Coolen et al., 1982] has quantification rank either one or dimensionality, measurement level nominal, bandwidth either one or two, and process discrete. Winsberg and Ramsay [1983] have, with some minor qualifications, measurement level ordinal, quantification rank unity, arbitrary bandwidth, and process discrete. PRINCIPALS [Young et al., 1978] has quantification rank one, measurement level nominal, ordinal or numerical, bandwidth either one or two, process continuous or discrete. But if the process is continuous the measurement level must be ordinal, and if the process is discreet the bandwidth must be one. It is clear that our new homogeneity analysis program, which only exists in preliminary APL-versions yet, encompasses all these possibilities and has all previous programs as special cases. Of course it will be more expensive in terms of time and storage, and more liable to produce degeneracy.

9. Words of Caution

Homogeneity analysis is a dangerous technique. We use very little information from the data, and we do not impose restrictions of a strong type on the representation. This type of program traditionally appeals greatly to many social scientists, who are very unsure about the value of their prior knowledge. They prefer to delegate the decisions to the computer, and they expect programs to generate knowledge. This strategy leads, all too often, to chance capitalization, triviality, and degeneracy. Hypotheses are never rejected, and investigators and constantly making errors of the second kind. As a consequence results can, of course, never be replicated. Generalized homogeneity analysis, as we have developed it here, is a very powerful tool which can contribute greatly to a further inflation of social science results. By choosing the least restrictive options we can make the results almost completely independent of the data.

On the other hand it is well known that if we pay too much attention to errors of the second kind, then social scientists can say absolutely nothing. This is also considered to be an undesirable state of affairs. It can be circumvented by concentrating on minute aspects of well-defined small problems, as in laboratory situations, or it can be circumvented by introducing vast quantities of prior knowledge, as in sociology. Of course in most cases the prior knowledge is nothing but prejudice, and it so dominates the investigation that the results become equally independent of the data.

This defines the dilemma of applied empirical social science. According to the canons of scientific respectability we can say almost nothing, and the things we can say are likely to be trivial. There are two ways out of this situation. Either we impose so much prior knowledge on our problem that the data only marginally make a difference. This is the rationalistic solution, popular in sociology. Or we impose so little prior knowledge that the data, including all outliers, stragglers, idiosyncracies, coding errors, missing data, completely determine the solution. In this case the technique is supposed to generate theory. This is the empiristic and technological approach, popular in applied psychology. Both approaches have, up to now, not produced much of interest.

Homogeneity analysis is firmly in the empiristic and technological tradition. Thus it is clear what dangers we have to guard against especially. If we have reliable prior knowledge, we must incorporate it. It is absolutely necessary to investigate the *stability* of the results [Gifi, 1981a; De Leeuw, 1984a]. Observe, however, that stability is not sufficient. A program that responds to any data matrix by drawing the unit circle is very stable indeed. We also need to *gauge* the technique, by comparing analysis with different options on data whose most important properties are known. For some forms of homogeneity analysis this has already be done quite extensively [Gifi, 1981a; De Leeuw, 1984c], but very little is known in this respect about the more general options discussed here. One strategy, that seems promising, is to analyze the same data which various options, and to see what is gained and what is lost if we switch from numerical to ordinal, from bandwidth one to bandwidth two, from discrete to continuous, and so on. In fact this defines another form of stability analysis, which seems indispensable in situations with little prior knowledge.

References

P. Bekker. Varianten van Niet-lineaire Principale Komponenten Analyse. Master's thesis, Department of Psychology, University of Leiden, 1982.

C. Burt. The Factorial Analysis of Qualitative Data. *British Journal of Statistical Psychology*, 3:166–185, 1950.

H. Coolen, J. van Rijckevorserl, and J. De Leeuw. An Algorithm for Nonlinear Principal Component Analysis with B-splines by Means of Alternating Least Squares. In H. Caussinus and P. Ettinger amd J.R. Mathieu, editors, *COMSTAT 1982*, volume II. Physika Verlag, Vienna, Austria, 1982.

J. De Leeuw. *Canonical Analysis of Categorical Data.* PhD thesis, University of Leiden, The Netherlands, 1973. Republished in 1985 by DSWO-Press, Leiden, The Netherlands.

J. De Leeuw. Nonlinear Principal Component Analysis. In H. Caussinus et al., editor, *COMPSTAT 1982*, pages 77–86, Vienna, Austria, 1982. Physika Verlag.

J. De Leeuw. Statistical Properties of Multiple Correspondence Analysis. Research Report RR-84-06, Department of Data Theory, University of Leiden, 1984a. URL `http://www.datatheory.nl/pdfs/84/84_06.pdf`.

J. De Leeuw. The Gifi System of Nonlinear Multivariate Analysis. In E. Diday, M. Jambu, L. Lebart, J. Pagès, and R. Tomassone, editors, *Data Analysis and Informatics*, volume III, Amsterdam, 1984b. North Holland Publishing Company.

J. De Leeuw. Models of Data. *Kwantitatieve Methoden*, 5:17–30, 1984c.

J. De Leeuw and P. Mair. Homogeneity Analysis in R: the Package `homals`. *Journal of Statistical Software*, (in press), 2009.

J. De Leeuw and J. Van Rijckevorsel. Beyond Homogeneity Analysis. In J.L.A. Van Rijckevorsel and J. De Leeuw, editors, *Component and Correspondence Analysis*. Wiley, Chichester, England, 1988.

J. De Leeuw and J. Walter. Optimal Scaling of Continuous Numerical Data. Technical report, Department of Data Theory, University of Leiden, 1977.

J. De Leeuw, F. W. Young, and Y. Takane. Additive Structure in Qualitative Data: an Alternating Least Squares Method with Optimal Scaling Features. *Psychometrika*, 41:471–504, 1976.

J. De Leeuw, J. Van Rijckevorsel, and H. Van Der Wouden. Nonlinear Principal Component Analysis Using B-Splines. *Methods of Operations Research*, 23:211–234, 1981.

A. Gifi. Nonlinear Multivariate Analysis. Technical report, Department of Data Theory, University of Leiden, 1981a.

A. Gifi. HOMALS Users Guide. Technical report, Department of Data Theory, 1981b.

A. Gifi. PRINCALS Users Guide. Technical report, Department of Data Theory, University of Leiden, 1982.

L. Guttman. The Quantification of a Class of Attributes: A Theory and Method of Scale Construction. In P. Horst, editor, *The Prediction of Personal Adjustment*, pages 321–348. Social Science Research Council, New York, 1941.

D. Lafaye de Michaux. *Approximations d'Analyses Canoniques Nonlineaires de Variables Aléatoires et Analyses Factorielles Priviléginates*. Thèse de Docteur-Ingenieur, Université de Nice, 1978.

J.F. Martin. Le Codage Flou et ses Applications en Statistique. Thèse de Troisieme Cycle, Université de Pau et des Pays de l'Adour, 1980a.

J.F. Martin. Une Approche des Codages Flous. Quelques Propriétes. Publications de Laboratoire de Statistique et Probabilité, Université Paul Sabatier, Toulouse, 1980b.

B.F. Schriever. Scaling of Order-dependent Categorical Variables with Correspondence Analysis. *International Statistical Review*, 51:225–238, 1983.

R.W. Tijssen. A New Approach to Nonlinear Canonical Correlation Analysis. Master's thesis, Department of Psychology, University of Leiden, 1985.

J. Van Rijckevorsel. Canonical Analysis with B-splibnes. In H. Caussinus, P. Ettinger, and R. Tomassone, editors, *COMPSTAT 1982*. Physika Verlag, Vienna, Austria, 1982.

S. Winsberg and J. O. Ramsay. Monotone Transformations to Additivity. *Biometrika*, 67:669–674, 1980.

S. Winsberg and J. O. Ramsay. Monotone Spline Transformations for Dimension Reduction. *Psychometrika*, 48:575–595, 1983.

F. W. Young. Quantitative Analysis of Qualitative Data. *Psychometrika*, 46:357–388, 1981.

F.W. Young, Y. Takane, and J. De Leeuw. The Principal Components of Mixed Measurement Level Multivariate Data: an Alternating Least Squares Method with Optimal Scaling Features. *Psychometrika*, 45: 279–281, 1978.

F.W. Young, J. De Leeuw, and Y. Takane. Quatifying Qualitative Data. In E.D. Lantermann and H. Feger, editors, *Similarity and Choice. Papers in Honor of Clyde Coombs*. Hans Huber, Bern, 1980.

Appendix A. Tables

	Price	Gas	Weight
Chevette	5.6	6.9	9.7
Dodge Colt	5.7	5.1	8.8
Plymouth Horizon	6.3	5.5	9.9
Fort Mustang	7.6	6.7	12.0
Pontiac Phoenix	8.6	6.9	12.1
Dodge Diplomat	9.4	10.2	15.5
Chevrolet Impala	10.1	7.5	16.9
Buick Regal	10.5	7.8	15.0
AMC Eagle	10.7	11.7	15.7
Oldsmobile 98	13.3	8.7	18.3

Table 1. Car Data

	Price	Gas	Weight
Chevette	1	1	1
Dodge Colt	1	1	1
Plymouth Horizon	1	1	1
Fort Mustang	2	1	2
Pontiac Phoenix	2	1	2
Dodge Diplomat	2	3	2
Chevrolet Impala	3	2	3
Buick Regal	3	2	2
AMC Eagle	3	3	2
Oldsmobile 98	4	2	3

Table 2. Car Data, Discrete

	Price	Gas	Weight
Chevette	1	4	2
Dodge Colt	2	1	1
Plymouth Horizon	3	2	3
Fort Mustang	4	3	4
Pontiac Phoenix	5	5	5
Dodge Diplomat	6	9	7
Chevrolet Impala	7	6	9
Buick Regal	8	7	6
AMC Eagle	9	10	8
Oldsmobile 98	10	8	10

Table 3. Car Data, Ranked

	Price			Gas			Weight			
Chevette	0.88	0.12	0.00	0.62	0.38	0.00	0.06	0.94	0.00	0.00
Dodge Colt	0.86	0.14	0.00	0.98	0.02	0.00	0.24	0.76	0.00	0.00
Plymouth Horizon	0.74	0.26	0.00	0.90	0.10	0.00	0.02	0.98	0.00	0.00
Fort Mustang	0.48	0.52	0.00	0.66	0.34	0.00	0.00	0.60	0.40	0.00
Pontiac Phoenix	0.28	0.72	0.00	0.62	0.38	0.00	0.00	0.58	0.42	0.00
Dodge Diplomat	0.12	0.88	0.00	0.00	0.96	0.04	0.00	0.00	0.90	0.10
Chevrolet Impala	0.00	0.98	0.02	0.50	0.50	0.00	0.00	0.00	0.62	0.38
Buick Regal	0.00	0.90	0.10	0.44	0.56	0.00	0.00	0.00	1.00	0.00
AMC Eagle	0.00	0.86	0.14	0.00	0.66	0.34	0.00	0.00	0.86	0.14
Oldsmobile 96	0.00	0.34	0.66	0.26	0.74	0.00	0.00	0.00	0.34	0.66

TABLE 4. Piecewise linear coding car data

Appendix B. Figures

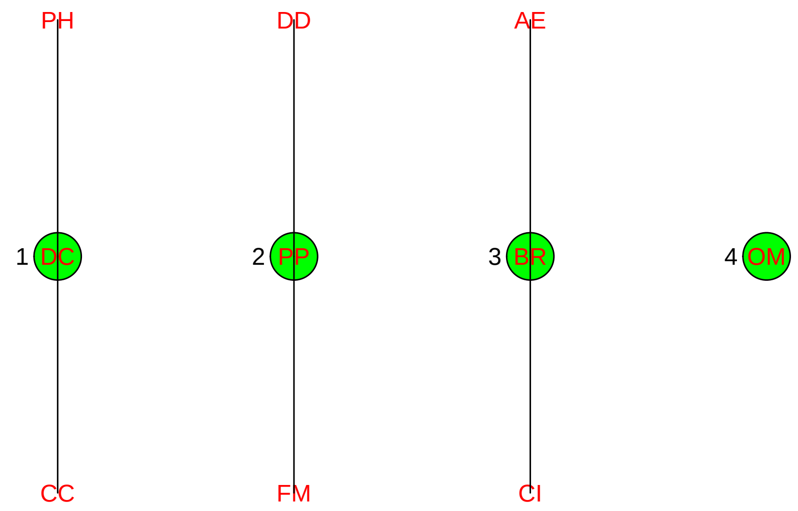

Figure 1. loss variable 1, arbitrary solution

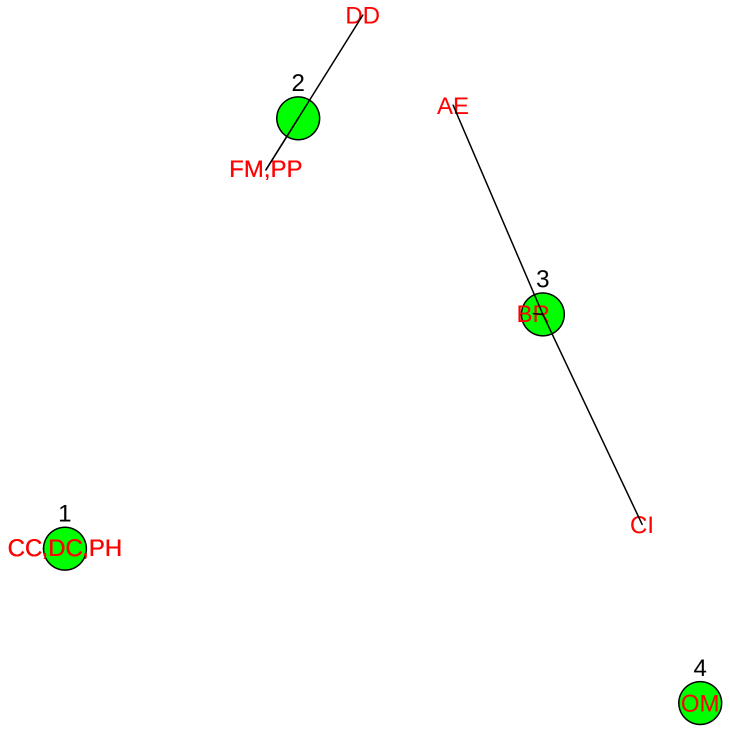

FIGURE 2. loss variable 1, optimal solution

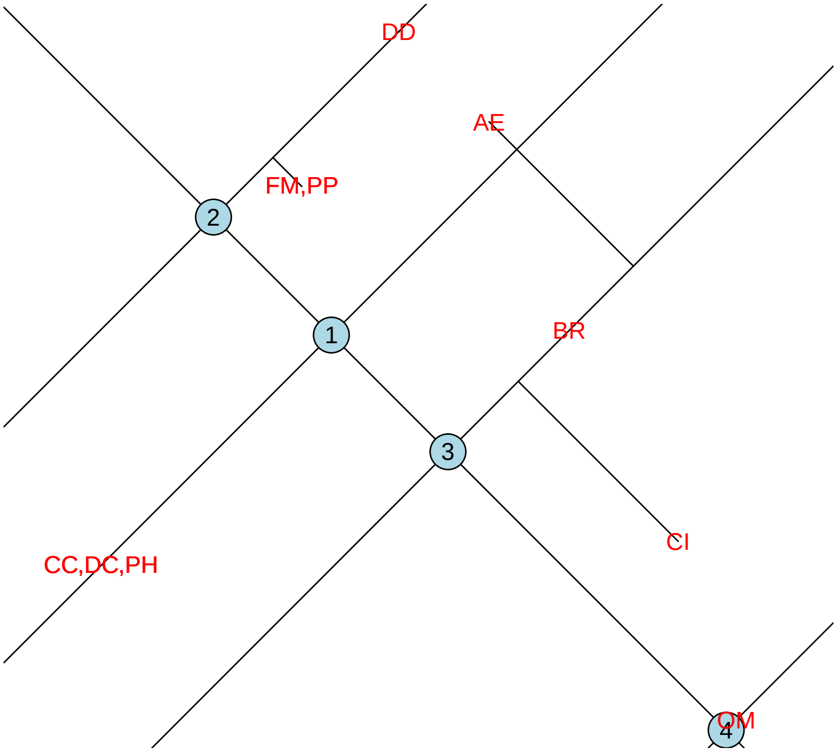

FIGURE 3. single nominal loss, variable 1, arbitrary solution

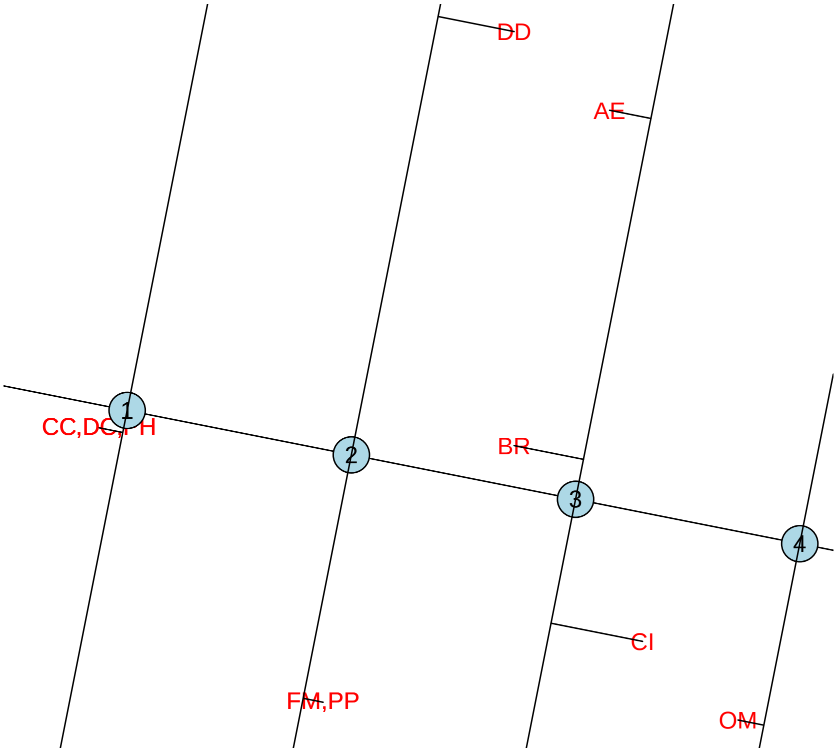

FIGURE 4. single numerical loss, variable 1, optimal solution

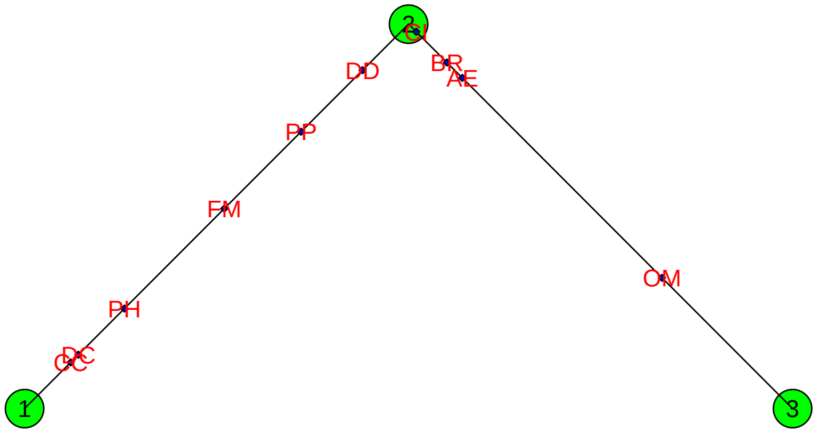

FIGURE 5. loss for multiple piecewise linear, variable1, arbitrary solution

Appendix C. Code

C.1. Figure 1.

```
x<-c(-1,-1,-1,0,0,0,1,1,1,2)
y<-c(-1,0,1,-1,0,1,-1,0,1,0)
xy<-cbind(x,y)
plot(xy,axes=FALSE,xlab="",ylab="",type="n",
        asp=1,xlim=c(-1.25,2))
symbols(-1,0,circles=.1,inches=FALSE,
        add=TRUE,fg="BLACK",bg="GREEN")
symbols(0,0,circles=.1,inches=FALSE,
        add=TRUE,fg="BLACK",bg="GREEN")
symbols(1,0,circles=.1,inches=FALSE,
        add=TRUE,fg="BLACK",bg="GREEN")
symbols(2,0,circles=.1,inches=FALSE,
        add=TRUE,fg="BLACK",bg="GREEN")
text(xy,c("CC","DC","PH","FM","PP","DD","CI","BR","AE","OM"),
        col="RED")
lines(matrix(c(-1,-1,-1,0),2,2,byrow=TRUE))
lines(matrix(c(-1,0,-1,1),2,2,byrow=TRUE))
lines(matrix(c(0,-1,0,0),2,2,byrow=TRUE))
lines(matrix(c(0,0,0,1),2,2,byrow=TRUE))
lines(matrix(c(1,-1,1,0),2,2,byrow=TRUE))
lines(matrix(c(1,0,1,1),2,2,byrow=TRUE))
text(matrix(c(-1.15,-.15,.85,1.85,0,0,0,0),4,2),
        c("1","2","3","4"))
```

C.2. Figure 2.

```
g1<-ifelse(outer(c(1,1,1,2,2,2,3,3,3,4),1:4,"=="),1,0)
g2<-ifelse(outer(c(1,1,1,1,1,3,2,2,3,2),1:3,"=="),1,0)
g3<-ifelse(outer(c(1,1,1,2,2,2,3,2,2,3),1:3,"=="),1,0)
p1<-g1%*%((1/colSums(g1))*t(g1))
p2<-g2%*%((1/colSums(g2))*t(g2))
p3<-g3%*%((1/colSums(g3))*t(g3))
xx<-eigen((p1+p2+p3)/3)$vectors[,c(2,3)]
xx[,1]<--xx[,1]
y1<-crossprod(g1,xx)/colSums(g1)
y2<-crossprod(g2,xx)/colSums(g2)
```

```
y3<-crossprod(g3,xx)/colSums(g3)
plot(xx,axes=FALSE,xlab="",ylab="",type="n",asp=1)
symbols(y1[1,1],y1[1,2],circles=.03,inches=FALSE,add=TRUE,fg="
    BLACK",bg="GREEN")
symbols(y1[2,1],y1[2,2],circles=.03,inches=FALSE,add=TRUE,fg="
    BLACK",bg="GREEN")
symbols(y1[3,1],y1[3,2],circles=.03,inches=FALSE,add=TRUE,fg="
    BLACK",bg="GREEN")
symbols(y1[4,1],y1[4,2],circles=.03,inches=FALSE,add=TRUE,fg="
    BLACK",bg="GREEN")
text(xx,c("CC,DC,PH","CC,DC,PH","CC,DC,PH","FM,PP","FM,PP","DD","
    CI","BR","AE","OM"),
        col="RED")
lines(matrix(c(xx[7,],y1[3,]),2,2,byrow=TRUE))
lines(matrix(c(xx[9,],y1[3,]),2,2,byrow=TRUE))
lines(matrix(c(xx[8,],y1[3,]),2,2,byrow=TRUE))
lines(matrix(c(xx[4,],y1[2,]),2,2,byrow=TRUE))
lines(matrix(c(xx[5,],y1[2,]),2,2,byrow=TRUE))
lines(matrix(c(xx[6,],y1[2,]),2,2,byrow=TRUE))
text(matrix(y1[1,]+c(0,.05),1,2),"1")
text(matrix(y1[2,]+c(0,.05),1,2),"2")
text(matrix(y1[3,]+c(0,.05),1,2),"3")
text(matrix(y1[4,]+c(0,.05),1,2),"4")
```

C.3. **Figure 3.**

```
g1<-ifelse(outer(c(1,1,1,2,2,2,3,3,3,4),1:4,"=="),1,0)
g2<-ifelse(outer(c(1,1,1,1,1,3,2,2,3,2),1:3,"=="),1,0)
g3<-ifelse(outer(c(1,1,1,2,2,2,3,2,2,3),1:3,"=="),1,0)
p1<-g1%*%((1/colSums(g1))*t(g1))
p2<-g2%*%((1/colSums(g2))*t(g2))
p3<-g3%*%((1/colSums(g3))*t(g3))
xx<-eigen((p1+p2+p3)/3)$vectors[,c(2,3)]
xx[,1]<--xx[,1]
d1<-colSums(g1)
y1<-crossprod(g1,xx)/d1
plot(xx,axes=FALSE,xlab="",ylab="",type="n",asp=1)
abline(0,-1)
b<-xx[c(1,6,8,10),2]-xx[c(1,6,8,10),1]
abline(b[1],1)
abline(b[2],1)
abline(b[3],1)
abline(b[4],1)
symbols(-b[1]/2,b[1]/2,circles=.025,inches=FALSE,
    add=TRUE,fg="BLACK",bg="LIGHTBLUE")
symbols(-b[2]/2,b[2]/2,circles=.025,inches=FALSE,
    add=TRUE,fg="BLACK",bg="LIGHTBLUE")
symbols(-b[3]/2,b[3]/2,circles=.025,inches=FALSE,
    add=TRUE,fg="BLACK",bg="LIGHTBLUE")
symbols(-b[4]/2,b[4]/2,circles=.025,inches=FALSE,
    add=TRUE,fg="BLACK",bg="LIGHTBLUE")
text(matrix(c(-b[1]/2,b[1]/2),1,2),"1")
text(matrix(c(-b[2]/2,b[2]/2),1,2),"2")
text(matrix(c(-b[3]/2,b[3]/2),1,2),"3")
text(matrix(c(-b[4]/2,b[4]/2),1,2),"4")
lines(matrix(c(xx[7,],(sum(xx[7,])-b[3])/2,(sum(xx[7,])+b[3])/2)
    ,2,2,byrow=TRUE))
lines(matrix(c(xx[9,],(sum(xx[9,])-b[3])/2,(sum(xx[9,])+b[3])/2)
    ,2,2,byrow=TRUE))
lines(matrix(c(xx[6,],(sum(xx[6,])-b[2])/2,(sum(xx[6,])+b[2])/2)
    ,2,2,byrow=TRUE))
lines(matrix(c(xx[4,],(sum(xx[4,])-b[2])/2,(sum(xx[4,])+b[2])/2)
    ,2,2,byrow=TRUE))
text(xx,c("CC,DC,PH","CC,DC,PH","CC,DC,PH","FM,PP","FM,PP","DD","
    CI","BR","AE","OM"),col="RED")
```

C.4. **Figure 4.**

```
h<-cbind(
   c(1,1,1,2,2,2,3,3,3,4),
   c(1,1,1,1,1,3,2,2,3,2),
   c(1,1,1,2,2,2,3,2,2,3))
lbs<-c("CC,DC,PH","CC,DC,PH","CC,DC,PH","FM,PP","FM,PP","DD","CI"
    ,"BR","AE","OM")
q<-scale(h)/3
s<-svd(q)
xx<-s$u[,c(1,2)]
a<-crossprod(q,x)[1,]
plot(xx,axes=FALSE,xlab="",ylab="",type="n",asp=1)
text(xx,lbs,col="RED")
```

```
abline(0,a[2]/a[1])
d<-c(3,3,3,1)
z<-1:4
z<-z-sum(d*z)/10
z<-z/sqrt(sum(d*z^2))
y<-outer(z,a)
symbols(y[1,1],y[1,2],circles=.025,inches=FALSE,
    add=TRUE,fg="BLACK",bg="LIGHTBLUE")
symbols(y[2,1],y[2,2],circles=.025,inches=FALSE,
    add=TRUE,fg="BLACK",bg="LIGHTBLUE")
symbols(y[3,1],y[3,2],circles=.025,inches=FALSE,
    add=TRUE,fg="BLACK",bg="LIGHTBLUE")
symbols(y[4,1],y[4,2],circles=.025,inches=FALSE,
    add=TRUE,fg="BLACK",bg="LIGHTBLUE")
text(matrix(c(y[1,1],y[1,2]),1,2),"1")
text(matrix(c(y[2,1],y[2,2]),1,2),"2")
text(matrix(c(y[3,1],y[3,2]),1,2),"3")
text(matrix(c(y[4,1],y[4,2]),1,2),"4")
abline(sum(y[1,]*a)/a[2],-a[1]/a[2])
abline(sum(y[2,]*a)/a[2],-a[1]/a[2])
abline(sum(y[3,]*a)/a[2],-a[1]/a[2])
abline(sum(y[4,]*a)/a[2],-a[1]/a[2])
r<-c(-a[2],a[1]); r<-r/sqrt(sum(r^2))
lines(rbind(xx[1,],y[1,]+sum((xx[1,]-y[1,])*r)*r))
lines(rbind(xx[4,],y[2,]+sum((xx[4,]-y[2,])*r)*r))
lines(rbind(xx[6,],y[2,]+sum((xx[6,]-y[2,])*r)*r))
lines(rbind(xx[7,],y[3,]+sum((xx[7,]-y[3,])*r)*r))
lines(rbind(xx[8,],y[3,]+sum((xx[8,]-y[3,])*r)*r))
lines(rbind(xx[9,],y[3,]+sum((xx[9,]-y[3,])*r)*r))
lines(rbind(xx[10,],y[4,]+sum((xx[10,]-y[4,])*r)*r))
```

C.5. **Figure 5.**

```
xx<-matrix(c(0,0,1,1,2,0),3,2,byrow=TRUE)
plot(xx,axes=FALSE,xlab="",ylab="",type="n",asp=1)
symbols(xx[1,1],xx[1,2],circles=.05,inches=FALSE,add=TRUE,fg="
    BLACK",bg="GREEN")
symbols(xx[2,1],xx[2,2],circles=.05,inches=FALSE,add=TRUE,fg="
    BLACK",bg="GREEN")
```

```
symbols(xx[3,1],xx[3,2],circles=.05,inches=FALSE,add=TRUE,fg="BLACK",bg="GREEN")
text(xx,c("1","2","3"))
lines(matrix(c(xx[1,],xx[2,]),2,2,byrow=TRUE))
lines(matrix(c(xx[2,],xx[3,]),2,2,byrow=TRUE))
g<-matrix(
c(
.88,.12,0,
.86,.14,0,
.74,.26,0,
.48,.52,0,
.28,.72,0,
.12,.88,0,
0,.98,.02,
0,.90,.10,
0,.86,.14,
0,.34,.66),10,3,byrow=TRUE)
y<-g%*%xx
points(y,pch=23,bg="BLUE",cex=.5)
text(y,c("CC","DC","PH","FM","PP","DD","CI","BR","AE","OM"),col="RED")
```

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