

# DATA VISUALIZATION THROUGH GRAPH DRAWING

GEORGE MICHAILIDIS AND JAN DE LEEUW

ABSTRACT. In this paper the problem of visualizing data sets is considered. By representing the data as the adjacency matrix of an appropriately defined bipartite graph, the problem is transformed to one of graph drawing. A general graph drawing framework is introduced, the corresponding mathematical problem defined and an algorithmic approach for solving the necessary optimization problem discussed. The new approach is illustrated through several examples.

## 1. INTRODUCTION

Advances in data collection, computerization of transactions and breakthroughs in storage technology have allowed research and business organizations to collect increasingly large amounts of data. In order to extract useful information from such large data sets, a first step is to be able to *visualize* their global structure and identify patterns, trends and outliers.

Graphs are useful entities since they can represent relationships between sets of objects. They are used to model complex systems (e.g. computer and transportation networks, VLSI and Web site layouts, molecules, etc) and to visualize relationships (e.g. social networks, entity-relationship diagrams in database systems, etc). Graphs are also very interesting mathematical objects and a lot of attention has been paid to their properties. In many instances the right picture is the key to understanding. The various ways of visualizing a graph provide different insights, and often hidden relationships and interesting patterns are revealed. An increasing body of literature is considering the problem of how to draw a graph (see for instance the book by [6] on Graph Drawing, the proceedings of the annual conference on Graph Drawing, etc). Also, several problems in distance geometry and in graph theory have their origin in the problem of graph drawing in higher dimensional spaces. Of particular interest are the representation of data sets through graphs. This bridges the fields of multivariate statistics and graph drawing.

Despite the recent explosive growth of the graph drawing field, the various techniques proposed exhibit a high degree of arbitrariness, and more often than not lack a rigorous mathematical background. In this paper, we provide a rigorous mathematical framework of drawing graphs utilizing the information contained in the adjacency matrix of the underlying graph. At the core of our approach are various loss functions that measure the lack of fit of the resulting representation, that need to be optimized subject to a set of constraints that correspond to different drawing

representations. We then establish how the graph drawing problem encompasses problems in multivariate statistics. We develop a set of algorithms based on the theory of majorization to optimize the various loss functions and study their properties (existence of solution, convergence, etc). We demonstrate the usefulness of our approach through a series of examples.

## 2. DATA AS GRAPHS

**2.1. Graphs and the Adjacency Matrix.** In this paper we consider an undirected graph  $G = (V, E)$ , where  $V = \{v_1, v_2, \dots, v_n\}$  is the set of the  $n$  vertices and  $E \subset V \times V$  the set of edges. It is assumed that the graph  $G$  does not contain either self-loops or multiple edges between any pair of vertices. The set of edges can be represented in matrix form through the *adjacency matrix*  $A = \{a_{ij} | i, j = 1, \dots, n\}$ . Thus, vertices  $i, j \in G$  are connected if and only if  $a_{ij} > 0$ , otherwise  $a_{ij} = 0$ . If  $a_{ij} \in \{0, 1\}$  we are dealing with a *simple* graph, otherwise a *weighted* graph.

In the following example the graph representation of a familiar data structure from multivariate statistics is given. Consider the following dissimilarity matrix

$$\begin{bmatrix} 0 & & & & & \\ 7 & 0 & & & & \\ 2 & 3 & 0 & & & \\ 8 & 6 & 1 & 0 & & \\ 5 & 9 & 6 & 3 & 0 & \\ 3 & 1 & 2 & 5 & 12 & 0 \end{bmatrix}$$

It can be represented by a complete weighted graph on 6 vertices as shown in Figure 1.

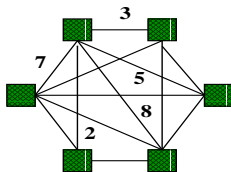


FIGURE 1. The graph representation of a dissimilarity matrix

**2.2. Bipartite Graphs.** In two recent papers of de Leeuw and Michailidis [5, 13] the problem of representing categorical datasets through *bipartite graphs* is considered. For bipartite graphs, the vertex set  $V$  is partitioned into two sets  $V_1$  and  $V_2$ , and the edge set  $E$  is defined on  $V_1 \times V_2$  and indicates which vertices from  $V_1$  are connected to vertices in  $V_2$  and vice versa. In multidimensional data analysis, the classical data structure where data on  $J$  categorical variables (with  $k_j$  possible values (categories) per variable) are collected for  $N$  objects, can be represented by a bipartite graph. The  $N$  objects correspond to the vertices of  $V_1$ , the  $K = \sum_j k_j$  categories to the vertices of  $V_2$  and there are  $N \times J$  edges in  $E$ , since each object is connected to  $J$  different categories.

The adjacency matrix of a bipartite graph takes the form

$$A = \begin{bmatrix} 0 & W \\ W' & 0 \end{bmatrix}$$

where  $W$  is the familiar superindicator matrix [8] from multiple correspondence analysis (MCA). The superindicator matrix of a small example (discussed in [18] and also [5]) and the corresponding graph representation are given next.

	cheap	not expensive	expensive	down fibers	synthetic fibers	good	acceptable	bad
Sleeping Bag	Price			Fiber		Quality		
One Kilo Bag	1	0	0	0	1	1	0	0
Sund	1	0	0	0	1	0	0	1
Kompakt Basic	1	0	0	0	1	1	0	0
Finmark Tour	1	0	0	0	1	0	0	1
Interlight Lyx	1	0	0	0	1	0	0	1
Kompakt	0	1	0	0	1	0	1	0
Touch the Cloud	0	1	0	0	1	0	1	0
Cat's Meow	0	1	0	0	1	1	0	0
Igloo Super	0	1	0	0	1	0	0	1
Donna	0	1	0	0	1	0	1	0
Tyin	0	1	0	0	1	0	1	0
Travellers Dream	0	1	0	1	0	1	0	0
Yeti Light	0	1	0	1	0	1	0	0
Climber	0	1	0	1	0	0	1	0
Viking	0	1	0	1	0	1	0	0
Eiger	0	0	1	1	0	0	1	0
Climber light	0	1	0	1	0	1	0	0
Cobra	0	0	1	1	0	1	0	0
Cobra Comfort	0	1	0	1	0	0	1	0
Foxfire	0	0	1	1	0	1	0	0
Mont Blanc	0	0	1	1	0	1	0	0

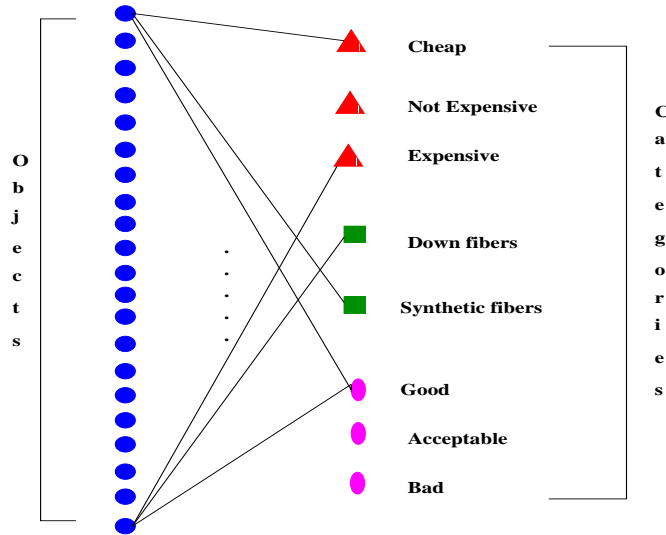


FIGURE 2. The bipartite graph of the sleeping bags example

Another data structure that can be represented by a bipartite graph is the contingency table, familiar from elementary statistics [1, 8], where the  $I$  categories of the first variable correspond to the vertices in  $V_1$  and the  $J$  categories of the second variable to those of  $V_2$ . For this data structure the  $a_{ij}$ 's are nonnegative numbers that indicate how many observations fall in cell  $(i, j)$  in the contingency table; thus, we are dealing with a weighted bipartite graph in this case. Finally, data sets involving measurements on objects organized in a hierarchical structure (e.g. students grouped by class or school, consumers grouped by geographical areas, etc) can be represented by direct sums of bipartite graphs [14].

### 3. GRAPH DRAWING

The adjacency matrix represents a useful way to code the information in a data set and moreover contains exactly the same information as the original data set; however, it is hard to use it to uncover patterns and trends in the data. One way of utilizing the information contained in the adjacency matrix is to draw the graph by connecting the appropriate vertices. This goes in the direction of making a picture of the data, and when things work out well, a picture is worth a lot of numbers, especially when these numbers are just zeros and ones. But the “technique” of drawing the coded graph, say in the plane, has a large amount of arbitrariness. Since the graph only contains the qualitative information of which vertices are connected, we can locate them anywhere in the plane and then draw the edges corresponding with the nonzero elements of the adjacency matrix. The trick is to manage to draw the graph in such a way so as the resulting picture becomes as informative as possible.

The general problem of graph drawing discussed in this paper is to represent the edges of a graph as points in  $\mathbb{R}^p$  and the vertices as lines connecting the points. Graph drawing is an active area in computer science, and it is very ably reviewed in the recent book by [6]. The choice of  $\mathbb{R}^p$  is due to its attractive underlying geometry and the fact that it renders the necessary computations more manageable.

There are basically two different approaches to make such drawings. In the *metric* or *embedding* approach we use the path-length distance defined between the vertices of the graph and we try to approximate these distance by the Euclidean distance between the points [2, 6].

In this paper, we adopt primarily the *adjacency model*, i.e. we do not emphasize graph-theoretical distance, but we pay special attention to which vertices are adjacent and which vertices are not. Obviously, this is related to distance, but the emphasis is different. We use objective (loss) functions to measure the *quality* of the resulting embedding.

**3.1. Force-Directed Techniques.** The class of graph drawing techniques we are most interested in here are the *force-directed techniques*. The vertices are bodies that attract and repel each other, for instance because the edges are springs or because the vertices have electric charges. This means that there are forces pulling and pushing the vertices apart, and the optimal graph drawing will be the one in which these forces are in equilibrium. In [6] (Chapter 10) force-directed graph

drawing means minimizing a loss function which incorporates both pushing and pulling. A popular choice for pulling is Hooke's law [7], i.e. the force is proportional to the difference between the distance of the vertices and the zero-energy length of the spring, while the repelling force follows an inverse square law. Force-directed algorithms are also employed in NicheWorks [17] a visualization tool for the investigation of very large graphs.

In this paper we concentrate on *pulling under constraints*, which means that we do not have explicit pushing components in the loss function. The constraints normalize the drawing in order to prevent trivial solutions in which the whole graph is pulled (collapses) into a single point.

Let us first define the following loss function, that represents in our study the main tool for making the necessary graph drawings,

$$(1) \quad \sigma_\phi(Z|A) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \phi(d_{ij}(Z)),$$

where  $d_{ij}(Z)$  denotes the distance of points with coordinates  $z_i$  and  $z_j$  in  $\mathbb{R}^p$ . We assume that the weights  $a_{ij} \in \{0,1\}$  and that  $\phi$  is an increasing function. Therefore, minimizing  $\sigma_\phi$  means minimizing the weighted sum of the transformed distances between the points that are connected in the graph. Notice that we do not assume that the distances are Euclidean, they could be  $\ell_1$  (City Block) or  $\ell_\infty$  (Chebyshev) or general  $\ell_p$  distances. Moreover, the presence of the  $\phi$  function allows us to mitigate the effect of the distances on the graphical representation we are seeking. For example, a convex  $\phi$  function will reinforce large distances by rendering them even larger, thus enabling us to detect unique features in the data. On the other hand, a concave function will dampen the effect of outlier objects (i.e. objects totally dissimilar to other objects in the data) in the final representation and thus provide us with a picture of the most important basic patterns in the data. Interesting choices of  $\phi$  functions include among others: the class of homogeneous power transformations given by  $\phi(d) = d^\gamma$ ,  $\gamma > 0$ , the logistic function  $\phi(d) = \exp(d)/(1 + \exp(d))$ , the logarithmic function  $\phi(d) = \log(d)$ , the Huber function  $\phi(d) = d^2/2$  for  $d < c$  and  $\phi(d) = cd - c/2$  for  $d > c$ , for some suitably chosen constant  $c > 0$ , etc. Also, the notation suggests that  $Z$  is the only variable that we control; for a given problem both  $\phi$  and  $A$  are fixed.

Minimizing the  $\sigma_\phi$  function without further restrictions does not make much sense. We can minimize it by simply collapsing all the points in the origin of the space ( $z_i = 0$  for all  $i$ ), and thus all corresponding distances become zero. This provides the global minimum of  $\sigma_\phi$ , but "Indeed, this is not a good drawing!" [6] (page 310). In order to avoid such trivial solutions we need to impose normalization restrictions. Our focus in this work is on imposing constraints on  $Z$ , so that the resulting graph drawing problem is well posed. Some of the possible normalizations are:

1. Tutte [19] suggests partitioning  $Z$  into (at least 3) fixed points  $X$  and free points  $Y$  and minimize  $\sigma_\phi$  over  $Y$  only.
2. Impose a constraint on  $Z$  (e.g.  $Z'Z = I$  or  $\text{trace}(Z'Z) = 1$ , etc).
3. Impose constraints on the distances (e.g.  $\sum_{ij} d^p(z_i, z_j) = 1$ ).

An alternative route would be to impose pushing constraints; i.e. vertices in the graph not connected are repelled.

It is important to realize that the choice of normalization is not a trivial matter. It will determine the shape and properties of the drawing. Thus it should be made in an informed way. It is also important that the data influence the drawing through  $\sigma_\phi$ , while the constraints are usually chosen by considerations of mathematical convenience or global properties of the drawing.

**3.2. The special case of squared Euclidean distances.** In this section we examine in more depth the special case where  $\phi = d_j^2(Z)/2$ ; that is, we examine the loss function with squared Euclidean distances. This turns out to be the most important case from the algorithmic point of view and many more general cases can be reduced to this one. The following notation is convenient: let  $d_{ij}^2(Z) = \text{trace}(Z'B_{ij}Z)$  with  $B_{ij} = (e_i - e_j)(e_i - e_j)'$  and where the  $e_i$ 's are unit vectors.

Define the matrix  $O$  by

$$(2) \quad O = \sum_{ij} a_{ij} B_{ij}.$$

Thus,  $O$  has the negative values  $-a_{ij}$  as its off-diagonal elements and the row-sums (or column-sums) of the adjacency matrix  $A$  as its diagonal elements. Thus,  $O$  is doubly-centered, and by construction positive semi-definite. We then have that

$$(3) \quad \sigma_2(Z|A) = \text{trace}(Z'OZ).$$

If in addition we choose  $Z'Z = I$  as the normalization, then the coordinates of the points in the optimal drawing correspond to the  $p$  eigenvectors corresponding to the smallest  $p$  non-zero eigenvalues. Thus, the solution to the problem becomes algorithmically straightforward.

On the other hand, if we decide to partition the coordinates of the solution  $Z = [X' Y']'$ , according to the two vertex sets (with  $X$  containing the coordinates of the vertices of the object points and  $Y$  those of the category points) and normalize only  $X$  (i.e.  $X'X = I$ ) or only  $Y$  (i.e.  $Y'Y = I$ ), then the loss function becomes identical to the one used in multiple correspondence analysis [8, 13]. In particular we have,

$$(4) \quad \sigma_2(X, Y|G) = \sum_{i=1}^N \sum_{k=1}^K g_{ik} d^2(x_i, y_k),$$

subject to the normalization constraint  $X'X = I$ , or equivalently in matrix form

$$(5) \quad \sigma_2(X, Y|G) = \text{trace}(JX'X + Y'DY - 2Y'GX),$$

where  $D = \text{diag}(G'G)$  and contains the univariate marginals of the categories and  $J$  corresponds to the number of variables in the data. There are two methods to minimize (5): (i) block relaxation and (ii) projection. In block relaxation the  $(X, Y)$  block structure is exploited. We alternate minimization over the variables in block  $Y$  by keeping the values in block  $X$  fixed, then minimize the variables in block  $X$  by keeping  $Y$  fixed and iterate between these two steps. This leads to the following well known alternating least squares algorithm [13]:

**Step 1:**  $\hat{Y} = D^{-1}G'X$

**Step 2:**  $\hat{X} = \frac{1}{J}GY$

**Step 3:** Orthonormalize  $X$  using the Gram-Schmidt procedure [9].

In projection methods, we have that for the optimal  $Y(X)$ , the  $\sigma_2$  function can be written as

$$(6) \quad \sigma_2(X|A) = \mathbf{trace}(X'(I - P)X),$$

where  $P = \frac{1}{J}GD^{-1}G'$  is the average *between categories* projector. Hence, the optimal  $X$  corresponds to the solution of the above eigenvalue problem. For both problems the solution at the optimum satisfies  $\hat{Y} = D^{-1}G'X$ ; i.e. category points are in the center of gravity of the objects belonging in a particular category. This is known in the literature as the *centroid principle* [8, 13].

It can be seen that the special case of squared Euclidean distances under the appropriate normalization recovers a well known multivariate technique.

*The Sleeping Bag example.* Using squared Euclidean distances to draw the graph of the sleeping bag example and using both the  $Z'Z = I$  and the  $X'X = I$  normalizations (only the coordinates of objects are normalized) we get the representations given in Figures 3 and 4, respectively.

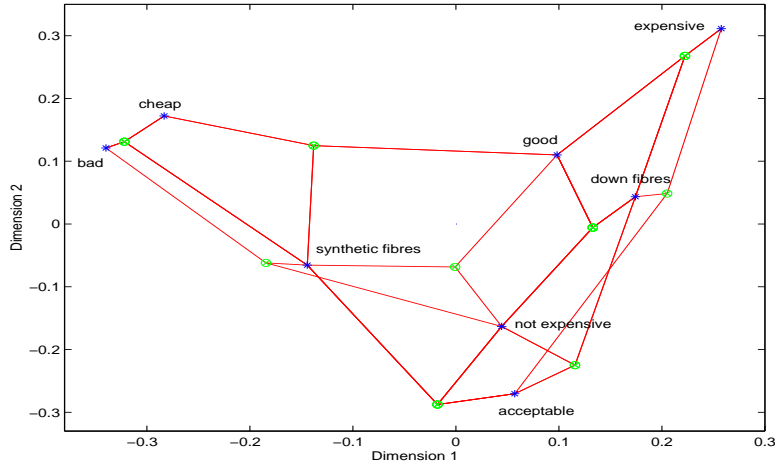


FIGURE 3. The graph drawing of the sleeping bags using Euclidean distances and the  $Z'Z = I$  normalization

The two representations reveal several things. First of all, the inherent rotational invariance of the solution. Moreover, both solutions capture the presence of good, expensive sleeping bags filled with down fibers and cheap, bad quality sleeping

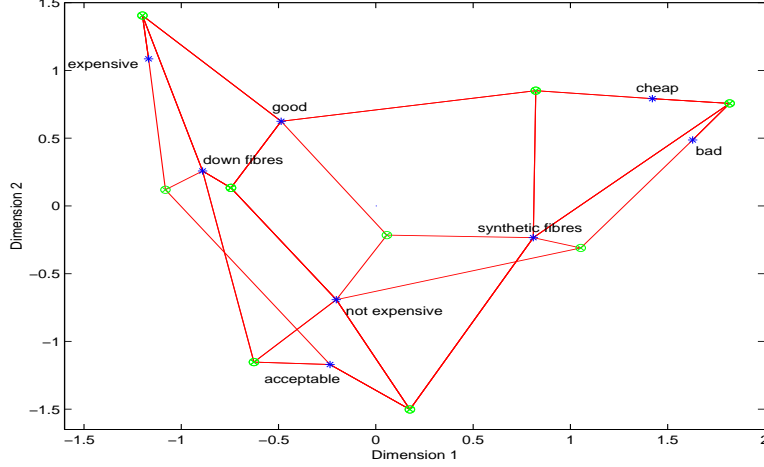


FIGURE 4. The graph drawing of the sleeping bags using Euclidean distances and the  $X'X = I$  normalization

bags filled with synthetic fibers and the absence of bad, expensive sleeping bags. It also shows that there are some intermediate sleeping bags in terms of quality and price filled either with down or synthetic fibers. However, the absence of a centroid principle for the representation based on the  $Z'Z = I$  normalization results in placing most vertices (both objects and categories) on the periphery of the graph. On the other hand, the presence of the centroid principle improves the representation aesthetically in Figure 4 and also serves as a guide to the data analyst for uncovering the basic patterns present in the data.

#### 4. A GENERAL OPTIMIZATION FRAMEWORK

One of the challenges that our general framework poses is the large number of possible configurations that we may encounter when optimizing the loss function  $\sigma_\phi$ . The different possibilities come about by the various combinations of parameters we can select, such as the type of distances (e.g.  $\ell_1$ ,  $\ell_2$ ,  $\ell_\infty$ ), the shape of the  $\phi$  function, the form of the normalization constraint. We propose a ubiquitous optimization scheme based on the principles of majorization that with few alterations can achieve our goal. Majorization is discussed in general terms in a series of papers [3, 4, 11, 12, 20].

Majorization algorithms use *surrogate* functions that render the optimization problem under consideration easier to handle. The goal (in general notation) is to optimize a function  $\zeta(\theta)$  over  $\theta \in \Theta$ , with  $\Theta \subseteq \mathcal{R}^p$ . Suppose that a function



$v(\theta, \chi)$  defined on  $\Theta \times \Theta$  satisfies

$$(7) \quad \zeta(\theta) \geq v(\theta, \chi) \text{ for all } \theta, \chi \in \Theta,$$

$$(8) \quad \zeta(\theta) = v(\theta, \theta) \text{ for all } \theta \in \Theta,$$

Thus, for a fixed  $\chi$ ,  $v(\bullet, \chi)$  is below  $\zeta$  and it touches  $\zeta$  at the point  $(\chi, \zeta(\chi))$ . We then say that  $\zeta(\theta)$  *majorizes*  $v(\theta, \chi)$  or that  $v(\theta, \chi)$  *minorizes*  $\zeta(\theta)$ . Two key results in [3] show that: (i) if  $\zeta$  attains its maximum on  $\Theta$  at  $\hat{\theta}$ , then  $v(\bullet, \hat{\theta})$  also attains its maximum of  $\Theta$  at  $\hat{\theta}$ , and (ii) if  $\tilde{\theta} \in \Theta$  and  $\hat{\theta}$  maximizes  $v(\bullet, \tilde{\theta})$  over  $\Theta$ , then  $\zeta(\hat{\theta}) \geq \zeta(\tilde{\theta})$ . These two results suggest the following algorithm for maximizing  $\zeta(\theta)$ .

- Step 1: Given a value  $\theta^{(k)}$  construct a minorizing function  $v(\theta^{(k)}, \chi)$ .  
 Step 2: Maximize  $v(\theta^{(k)}, \chi)$  with respect to  $\chi$  and set  $\theta^{(k+1)} = \chi^{\max}$ .  
 Step 3: If  $|\zeta(\theta^{(k+1)}) - \zeta(\theta^{(k)})| < \epsilon$  for some predetermined small  $\epsilon$  stop; else, go to Step 1.

In order for this algorithm to be of practical use, the minorizing function  $v$  should be easily maximized. Obviously, in case we are interested in minimizing  $\zeta$  a majorizing function  $v$  that needs to be minimized in Step 2 needs to be identified.

For the problem at hand it can be shown that (details given in [15]) for almost all choices of transformations that we are interested in, the loss function  $\sigma_\phi(Z|A)$  is majorized by  $\text{trace}(Z'C(Z, A)Z)$  for an appropriately defined matrix  $C(Z, A)$ . This latter fact implies that under an orthonormality normalization constraint, at every step of the algorithm we just have to solve an eigenvalue problem.

For example, suppose that we are interested in using  $\phi = d^\beta$  with  $\beta = [1, 2]$ . This is a convex function with growth rate slower than the quadratic. It contains as special cases both the  $\sigma_2$  function and the  $\sigma_1$  function that deals with Euclidean distances (distances without the square). The corresponding minimization problem can be easily solved by constructing a majorization function based on Young's inequality.

$$(9) \quad d_{ij}^\beta(Z) \leq \frac{2-\beta}{2} d_{ij}^\beta(\tilde{Z}) + \frac{2}{\beta d_{ij}^{2-\beta}(\tilde{Z})} d_{ij}^2(Z),$$

which implies that we can construct a quadratic majorizing function, and hence get

$$\begin{aligned} \sigma_\beta(Z) \leq \sum_{i=1}^n \sum_{j=1}^n a_{ij} \left( \frac{2-\beta}{2} d_{ij}^\beta(\tilde{Z}) + \frac{2}{\beta d_{ij}^{2-\beta}(\tilde{Z})} d_{ij}^2(Z) \right) = \\ \text{trace} \left( \frac{2-\beta}{2} \tilde{Z}' C_\beta(\tilde{Z}, A) Z + \frac{2}{\beta} Z' C_\beta(\tilde{Z}, A) Z \right), \end{aligned}$$

where

$$(10) \quad C_\beta(\tilde{Z}, A) = \sum_{i=1}^n \sum_{j=1}^n \frac{a_{ij}}{d_{ij}^{2-\beta}(\tilde{Z})} A_{ij}, \quad \beta \in [1, 2].$$

Thus, in an iteration we minimize  $\text{trace}(Z'C_\beta(Z^{\text{previous}}, A)Z)$  over normalized  $Z$ , where  $Z^{\text{previous}}$  denotes the value of  $Z$  in the previous iteration.

## 5. BIPARTITE GRAPHS AND PARALLEL COORDINATE PLOTS

In this section it is shown how our graph drawing framework can help improve the presentation of multivariate categorical data using parallel coordinates plots [21]. In such plots, we draw  $J$  parallel straight lines, one for each variable. The objects are then plotted on each of the lines, and points corresponding with the same objects are connected by broken line segments (and perhaps colored with different colors). These plots (implemented in many data visualization packages) capture bivariate relationships and can reveal various interesting features in the data. However, the parallel coordinate plot for nominal variables may not be particularly illuminating. In this section we use the mammals dentition data set for illustration purposes (the data are analyzed in [13]). A short description of the variables and their frequencies are given in Appendix A. The parallel coordinates plot of this data set from XGobi is given in Figure 5. The question is whether some other arrangement

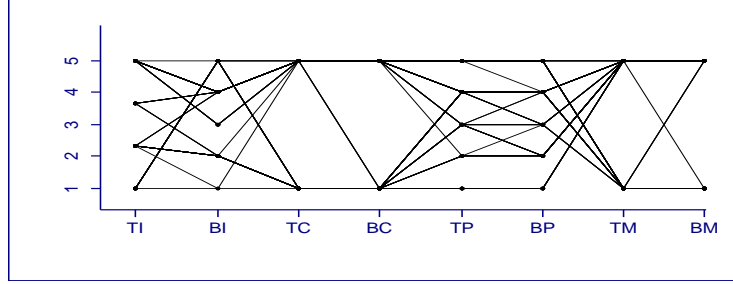


FIGURE 5. Parallel coordinates plot of the mammals data set

of the category points would reduce the number of crossings and thus lead to a cleaner and more informative representation of the data. Suppose that we are allowed to place the category points of variables  $j$  at arbitrary locations along the  $j$ th vertical lines. We propose to use the coordinates for the categories found by the multiple correspondence analysis solution. Notice that each object defines a line through  $J$  category points. Let  $q_{ij}$  be the induced quantification of object  $i$  on variable  $j$ , which is the same value as the corresponding quantification of the category of variable  $j$  that object  $i$  belongs in. We can partition the variance in the induced quantifications, as in the following table 1

Source	Sum of Squares	Matrix Expression
Within Objects, Between Variables	$\sum_{i=1}^n \sum_{j=1}^J (q_{ij} - q_{i\bullet})^2$	$y'(D - \frac{1}{J}C)y$
Between Objects	$J \sum_{i=1}^n (q_{i\bullet} - q_{\bullet\bullet})^2$	$\frac{1}{J}y'Cy$
Total Variance	$\sum_{i=1}^n \sum_{j=1}^J (q_{ij} - q_{\bullet\bullet})^2$	$y'Dy$

TABLE 1. Partitioning Quantification Variance

where  $C = G'G$ ,  $D = \text{diag}(G'G)$  and  $y$  is a column vector containing the quantifications of the category points of all the variables. This measures in how far the lines connecting the objects deviate from horizontal lines, by computing the variance around the best fitting horizontal line, which is given by  $q_{i\bullet}$ . Minimizing the ratio

of the within-object variance  $y'(D - \frac{1}{J}C)y$  to the total variance  $y'Dy$  amounts to computing the first dimension of an MCA. Observe that this is the same as maximizing the between-object variance  $\frac{1}{J}y'Cy$  for a given total variance, i.e. we also want the horizontal lines to be as far apart as possible. This is discussed in more detail in Chapter 3 in [8]. We illustrate the above with our mammals example. It can be seen that in the layout shown in Figure 6 the number of crossings is significantly reduced compared to the original plot (20 crossings vs 30). Moreover, the basic classification of the mammals is uncovered.

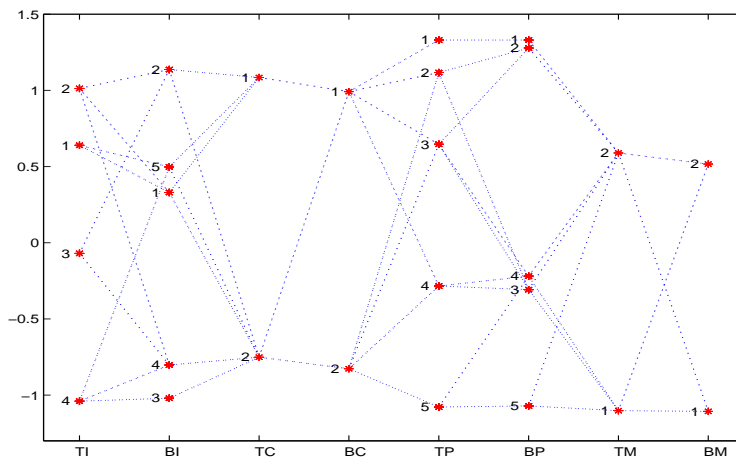


FIGURE 6. Optimized parallel coordinates plots for the mammals data

## 6. CHOICE OF PARAMETERS

In this section we briefly look at the impact of the choice of some of the parameters in our framework such as the normalization constraint and the transformation function on the resulting representation. In Figures 7 and 8 the graph layouts of the mammals dentition data set is given under the  $\sigma_2$  and  $\sigma_{1.6}$  loss functions, respectively.

It can be seen that in the latter case the graph drawing exhibits a much stronger clustering pattern of both the object and category points. This suggests that for large data sets such a solution could focus on the most basic patterns present in the data and ignore the minor ones. However, one should be careful, since under the  $\sigma_1$  function the resulting representation consists of  $p + 1$  points ( $p$  being the embedding dimension) and thus leads to a rather trivial solution. As argued in [16] this is due mostly to the choice of an orthonormality constraint.

On the other hand, the Tutte normalization heavily depends on the choice of the points being fixed as the Figures 9-11 indicate. For example, fixing the category points of a single variable leads to a representation where all the other points are clumped near the center of the plot (e.g. Figures 9 and 10), while fixing the

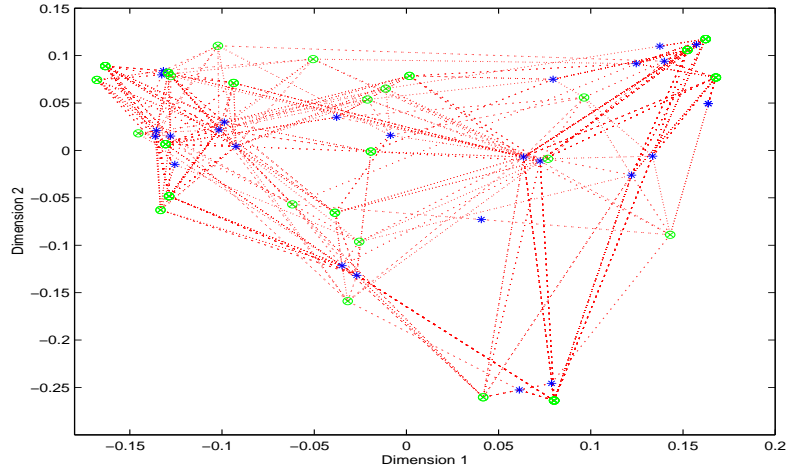


FIGURE 7. The graph representation of the mammals data using the  $\sigma_2$  loss function

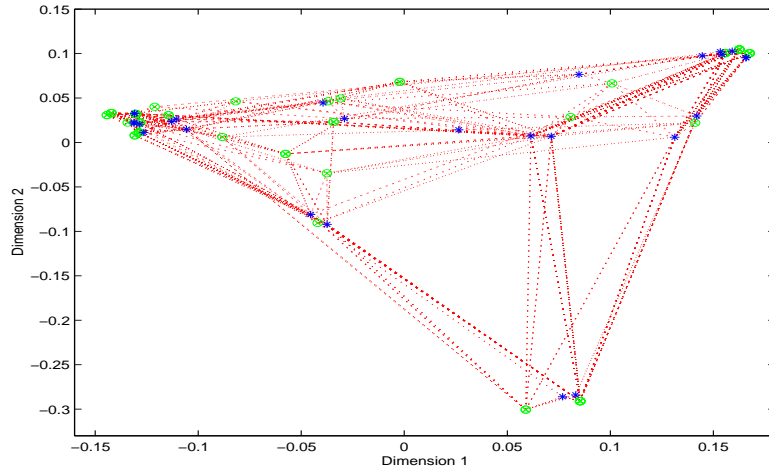


FIGURE 8. The graph representation of the mammals data using the  $\sigma_{1.6}$  loss function

coordinates of all the points of one of the vertex sets can lead to informative plots (e.g. Figure 11).

## 7. CONCLUDING REMARKS

In this study a general framework of visualizing data sets through graph drawing is considered. The underlying mathematical problem is rigorously defined and a

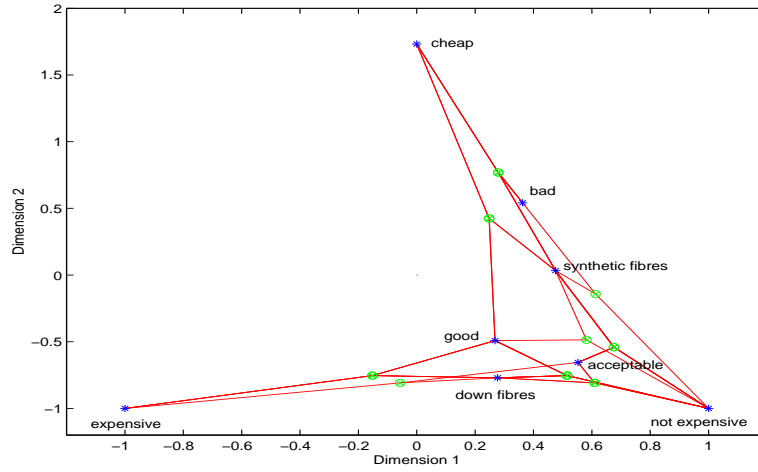


FIGURE 9. Tutte solution with the points corresponding to variable Price fixed

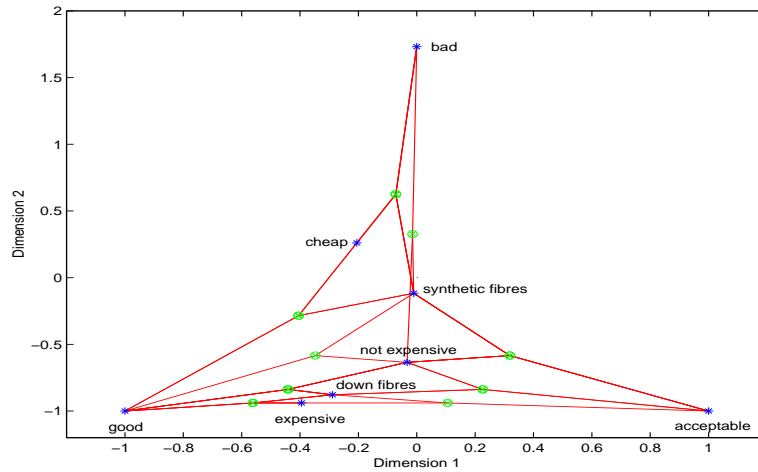


FIGURE 10. Tutte solution with the points corresponding to variable Quality fixed

ubiquitous optimization framework based on the concept of majorization is introduced. It is further shown that the familiar technique of Multiple Correspondence Analysis (MCA) is a special case of our general framework. The MCA solution can also be used to improve the presentation of categorical data sets through parallel coordinates plots. Moreover, the choice of normalizations, as well as the type of distances and the transformation function  $\phi$  have a significant impact on the resulting representation. An important future direction is the introduction of pushing constraints in the framework and the analysis of weighted graphs that represent data sets with numerical variables.

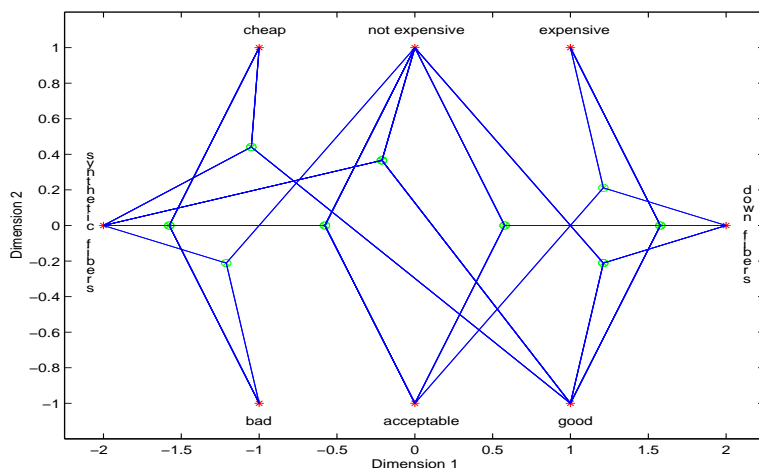


FIGURE 11. Tutte solution with the points corresponding to all the variables fixed

## 8. APPENDIX A - DENTITION OF MAMMALS

The data for this example are taken from Hartigan's book [10]. Mammals' teeth are divided into four groups: incisors, canines, premolars and molars. A description of the variables with their respective coding is given next.

- TI:** Top incisors; 1: 0 incisors, 2: 1 incisor, 3: 2 incisors, 4: 3 or more incisors
- BI:** Bottom incisors; 1 : 0 incisors, 2: 1 incisor, 3: 2 incisors, 4: 3 incisors, 5: 4 incisors
- TC:** Top canine; 1: 0 canines, 2: 1 canine
- BC:** Bottom canine; 1: 0 canines, 2: 1 canine
- TP:** Top premolar; 1: 0 premolars, 2: 1 premolar, 3: 2 premolars, 3: 2 premolars, 4: 3 premolars, 5: 4 premolars
- BP:** Bottom premolar; 1: 0 premolars, 2: 1 premolar, 3: 2 premolars, 3: 2 premolars, 4: 3 premolars, 5: 4 premolars
- TM:** Top molar; 1: 0-2 molars, 2: more than 2 molars
- BM:** Bottom molar; 1: 0-2 molars, 2: more than 2 molars

In the following table the frequencies of the variables are given.

The complete data set is given in [13].

Variable	Categories				
	1	2	3	4	5
TI	15.2	31.8	13.6	39.4	15.2
BI	3.0	30.3	7.6	43.9	
TC	40.9	59.1			
BC	45.5	54.5			
TP	9.1	10.6	18.2	39.4	22.7
BP	9.1	18.2	15.2	36.4	21.2
TM	34.8	65.2			
BM	31.8	68.2			

TABLE 2. Mammals teeth profiles (in %, N=66)

## REFERENCES

- [1] Benzécri, J.P. (1992), *Correspondence Analysis Handbook*, Marcel Dekker, Inc.
- [2] Buja, A., Swayne, D.F., Littman, M. & Dean, N. (1998), 'XGvis: Interactive data visualization with multidimensional scaling', under review *Journal of Computational and Graphical Statistics*
- [3] De Leeuw, J. (1994), 'Block relaxation algorithms in statistics', *Information Systems and Data Analysis*, Bock et al. (eds), Springer
- [4] De Leeuw, J. & Michailidis, G. (2000), 'Majorization methods in statistics', *Journal of Computational and Graphical Statistics*, **9**, 26-31
- [5] De Leeuw, J. & Michailidis, G. (2000), 'Graph layout techniques and multidimensional data analysis', *Game Theory, Optimal Stopping, Probability and Statistics. Papers in honor of Thomas S. Ferguson*, F.T. Bruss and L. Le Cam (eds), IMS Lecture Notes-Monograph Series, 219-248
- [6] Di Battista, G., Eades, P., Tammasia, R. & Tollis, I. (1998), *Graph Drawing: Algorithms for the Geometric Representation of Graphs*, Prentice Hall
- [7] Eades, P. (1984), 'A heuristic for graph drawing', *Congressus Numerantium*, **42**, 149-160
- [8] Gifi, A. (1990), *Nonlinear Multivariate Analysis*, Wiley
- [9] Golub, G.H. & Van Loan, C. (1997), *Matrix Computations*, Johns Hopkins University Press (3rd ed)
- [10] Hartigan, J. (1975), *Clustering Algorithms*, Wiley
- [11] Heiser, W.J. (1995), 'Convergent computing by iterative majorization: theory and applications in multidimensional data analysis', in *Recent Advances in Descriptive Multivariate Analysis*, Krzasnowski (ed.), Clenredon Press
- [12] Lange, K., Hunter, D.R. & Yang, I (2000), 'Optimization transfer using surrogate objective functions', *Journal of Computational and Graphical Statistics*, **9**, 1-25
- [13] Michailidis, G. & de Leeuw, J. (1998), 'The Gifi system for descriptive multivariate analysis', *Statistical Science*, **13**, 307-336
- [14] Michailidis, G. & de Leeuw, J. (2000), 'Multilevel homogeneity analysis with differential weighting', *Computational Statistics and Data Analysis*, **32**, 411-442
- [15] Michailidis, G. & de Leeuw, J. (2000), 'Multivariate data analysis using constrained pulling', Technical report, Department of Statistics, The University of Michigan
- [16] Michailidis, G. & de Leeuw, J. (2000), 'Homogeneity analysis by alternating least absolute deviations', Technical report, Department of Statistics, The University of Michigan
- [17] Wills, G.J. (1999), 'NicheWorks: Interactive visualization of very large graphs', *Journal of Computational and Graphical Statistics*, **8**, 190-212
- [18] Prediger, S. (1997), 'Symbolic objects in formal concept analysis', Proceedings of the Second International Symposium on Knowledge, Retrieval, Use and Storage for Efficiency, G. Mineau & A. Fall (eds)
- [19] Tutte, W.T. (1963), 'How to draw a graph', *Proceedings of the London Mathematical Society*, **13**, 743-767

- [20] Verboon, P. (1994), *A Robust Approach to Nonlinear Multivariate Analysis*, DSWO Press
- [21] Wegman, E.J. (1990), 'Hyperdimensional data analysis using parallel coordinates', *Journal of the American Statistical Association*, **85**, 664-675

DEPARTMENT OF STATISTICS, THE UNIVERSITY OF MICHIGAN, ANN ARBOR

*E-mail address*, George Michailidis: `gmichail@umich.edu`

DEPARTMENT OF STATISTICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES

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