DRAWING DATA GRAPHS BY PUSHING AND PULLING

JAN DE LEEUW AND GEORGE MICHAILIDIS

ABSTRACT. In this paper we discuss three different disciplines: graph drawing, multivariate descriptive statistics and location analysis. These areas share a common core of loss functions and algorithms, in which functions of distances are optimized by majorization methods. We discuss and extend these common components.

CONTENTS

Date: August 4, 1999.

1. INTRODUCTION

Graphs are useful entities since they can represent relationships between sets of objects. They are used to model complex systems (e.g. transportation networks, VLSI layouts, molecules etc) and to visualize relationships (e.g. social networks). 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 Tammasia's book 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 mutlivariate statistics and graph drawing. Moreover, a field in operations research with a long history is location analysis. The goal is to optimally place a new set of facilities that maximize some reward function subject to demand constraints (using the language of this field). It is shown later on that the basic problems in location analysis can be cast after some appropriate transformations to a graph drawing problem.

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 and location analysis. 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. Finally, we examine some special situations (gauges) and show how our techniques recover correctly the underlying graph structure.

2. DATA AS GRAPHS

2.1. **Graphs and the Adjacency Matrix.** Consider an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ 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* $W = \{w_{ij}|,\ i, j = 1, \ldots, n\}$. Thus, vertices $i,\ j \in G$ are connected if and only if $w_{ij} > 0$, otherwise $w_{ij} = 0$. In case all w_{ij}'s are 0 or 1 we have a *simple* graph, otherwise a *weighted graph*.

In a simple graph $w_{ij} = 1$ indicates that edges i and j are connected. Consequently, in a weighted graph, it makes sense to interprete a large w_{ij} to indicated a large "degree of connectedness", or a large *similarity* of the edges.

2.2. **Bipartite Graphs.** In Michailidis and de Leeuw [1999a] and de Leeuw and Michailidis [1999] the problem of representing datasets with categorical variables as *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 $V_1 \otimes V_2$ 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 [Michailidis and de Leeuw, 1999a, de Leeuw and Michailidis, 1999]. 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 w_{ij}'s take values in $\{0, 1\}$, and hence we are dealing with a simple bipartite graph.

Another data structure that can be represented by a bipartite graph is the contingency table, familiar from elementary statistics [Gifi, 1990, Benzécri, 1992], where the I categories of the first variable correspond to the vertices in V_1 and the L categories of the second variable to those of V_2. For this data structure the w_{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.

2.3. **Sums of Bipartite Graphs.** In many situations objects are naturally clustered into groups. For example, in educational research students are grouped by class or school, in sociological research individuals are grouped by socioeconomic status, in marketing research consumers are clustered in geographical regions, while in longitudinal studies we have repeated measurements on individuals. In the first example groups correspond to classes or schools, in the second to various a priori defined levels of socioeconomic status, in the third to regions (such as counties, states or even the northeast, the southwest etc), and in the fourth example to time periods. The data structure for each group corresponds to a bipartite graph and that of the entire data set to a *direct sum* of bipartite graphs. Different ways of analyzing such data are given in Michailidis and de Leeuw [1997, 1999b].

2.4. **Multipartite Graphs.** For multipartite graphs, the vertex set V is partitioned into M sets $V_1, \ldots, V_M$, and the edge set E is defined on $V_m \otimes V_{m'}$, $m, m' \in \{1, \ldots, M\}$. Multipartite graphs can be used to represent the data structure of relational data sets. Consider the following situation. There are two sets of objects, each one characterized by a set of attributes. For example in a database, we may have individuals described by a set of attributes such as different types of assets they own, income, occupation, other background characteristics, etc, and financial institutions described by a different set of attributes, such as profits, products they offer, etc. Moreoever, the two sets of objects, namely individuals and financial institutions, are related to each other, since individuals may do business (e.g. mortgage loans, credit cards) with several institutions and institutions may have as clients several individuals. In this example, the four vertex sets correspond to the individuals and their attributes, and to the financial institutions and their attributes and the edges link the two sets of objects and the objects with their attributes. However, there are no edges linking the financial institutions to their clients' attributes, or the individuals with the institutions' attributes. More complicated relational databases give rise to multipartite graphs with a large number of vertex sets.

2.5. **Complete Weighted Graphs.** This type of graphs is comprised of a set of N vertices and $\binom{N}{2}$ edges; i.e. each vertex is connected to all other $N-1$ vertices. Moreover, each edge has a nonnegative weight; hence, in this case $W = \{w_{ij}\}$ is a symmetric matrix with zeros along the main diagonal. Such graphs arise in multidimensional scaling (MDS) where the weights (w_{ij}) correspond to measures of similarity between the N objects (the vertices of the graph). In fact, in MDS measures of dissimilarity are more commonly used than measures of similarity, but we have seen above that in our basic interpretation large weights mean adjacency or closeness.

2.6. **Function and Regression Graphs.** Remember that the graph of a function $f : X \to Y$ is a subset of $X \otimes Y$. This means we can think of the graph of a function as a special bipartite graph, in which each element of V_1 is connected with exactly one element of V_2. Thus the rows of the adjacency matrix add up to one.

The categorical variables indicator graphs of Section 2.2 are columnwise direct sums of J of these function graphs, where all functions are defined on the set of N objects.

3. Graph Drawing

In the previous section we have indicated that many forms of data can be coded as adjacency matrices of simple or weighted graphs. This is a

useful way to think about coding, but the adjancency matrix of the graph is not a particularly nice representation of the data. We can, of course, also draw the graph and connect 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 zeroes and ones. But the "technique" is drawing the coded graph, say in the plane, has a enormous amount of arbitraryness. Since the graph only contains the qualitative information 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 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 di Battista et al. [1998].

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 edges of the graph and we try to approximate these distance by the Euclidean distance between the points [di Battista et al., 1998, section 10.3]. The area of embedding graph-theoretical distances is related to distance geometry, and it has been studied a great deal recently. For a review, see Michailidis and de Leeuw [1999c].

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 slightly different. through objective functions that 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 repell each other, for instance because the edges are springs or because the vertices have electric charges. This means that there are forces polling and pushing the vertices apart, and the optimal graph drawing will be the one in which these forces are in equilibrium.

In di Battista et al. [1998, Chapter 10] force-directed graph drawing means minimizing a loss function which incorporates both pushing and pulling. di Battista et al. [1998] propose that the pulling is done by springs obeying Hooke's law, i.e. the force is proportional to the difference between the distnace between the vertices and the zero-energy length of the spring. The electrical force that pushes follows an inverse square law.

In order to make this more general, we first define

$$\textbf{pull}_\phi(Z) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\phi(d_{ij}(Z)), \tag{1}$$

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 w_{ij} are non-negative and that ϕ is an increasing function. Therefore, minimizing **pull** means minimizing the weighted sum of the transformed distances between the points that are connected in the graph[1].

In the proposal of di Battista et al. [1998] we have

$$\phi(d_{ij}(Z)) = \begin{cases} w_{ij}(d_{ij}(Z) - \ell_{ij})^2 & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise .} \end{cases} \tag{2}$$

Here w_{ij} is the *stiffness* of the spring, and ℓ_{ij} is its natural or zero-energy length. Obviously, in this case ϕ is not increasing. Thus (2) is not completely in our framework, but we shall see later how to fit it in.

In an earlier proposal by Tutte [1963] the zero-energy lengths are all chosen equal to zero and the stiffnesses are all set equal to one. Thus

$$\textbf{pull}_{Tutte}(Z) = \sum_{(v_i,v_j)\in E} d_{ij}^2(Z). \tag{3}$$

Minimizing this pull 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. "Indeed, this is not a good drawing !" [di Battista et al., 1998, page 310].

We solve this problem by defining a second **pull** function with transformation ψ and weights u_{ij}, and then combining the two in

$$\textbf{pullpush}^{\mathcal{F}}_{\phi\psi}(Z) = \mathcal{F}(\textbf{pull}_\phi(Z), \textbf{pull}_\psi(Z)), \tag{4}$$

where $\mathcal{F}$ is increasing in its first argument and decreasing in its second argument. Two obvious choices are

$$\textbf{pullpush}^{-}_{\phi\psi}(Z) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\phi(d_{ij}(Z)) - \sum_{i=1}^{n}\sum_{j=1}^{n} u_{ij}\psi(d_{ij}(Z)). \tag{5a}$$

and

$$\textbf{pullpush}^{/}_{\phi\psi}(Z) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\phi(d_{ij}(Z))}{\sum_{i=1}^{n}\sum_{j=1}^{n} u_{ij}\psi(d_{ij}(Z))} \tag{5b}$$

[1]Observe we do not assume that the distances are Euclidean, they could be ℓ_1 (City Block) or ℓ_∞ (Chebyshev) distances.

The **pullpush** function can also be used if weights are not necessarily non-negative (such as in location theory, when locating obnoxious facilities). We can always write $w_{ij} = w_{ij}^+ - w_{ij}^-$, with both components non-negative, and if we substitute this in **pull** we get a special case of **pullpush**. In the same way, if ϕ is not increasing, we can write $\phi = \phi^+ - \phi^-$, with both components increasing. Using this in **pull** again gives a special case of **pullpush**. The **pull** function was first discussed in a general data analysis context by Heiser [1981]. Before that, it has been used extensively in location and assigment problems. The pushing and pulling terminology is quite common there, see for example Eiselt and Laporte [1995].

3.2. **Normalization.** Suppose we do not explicitly want to push, but we want to impose restrictions on the configuration Z in such a way that trivial solutions are avoided. As we have seen, Tutte [1963] partitions Z into fixed points X and free points Y. If the restrictions on the configuration can be formulated in terms of distances, then (5a) can be interpreted as a Lagrangian in which the weights u_{ij} are now Lagrange multipliers.

There is another way in which restrictions can be incorporated. If ψ is homogeneous (of any degree), then minimizing (5b) is the same as minimizing $\mathbf{pull}_\phi(Z)$ under the condition that $\mathbf{pull}_\psi(Z) = 1$. An important class of homogeneous pull functions is

$$\mathbf{pull}_\sigma(Z) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} d_{ij}^\sigma(Z), \tag{6}$$

in which we look at sum of (not necessarily integral) powers of the distances. The special cases $\mathbf{pull}_1$ and $\mathbf{pull}_2$ have been most closely studied.

4. Location and Assignment Problems

4.1. **The Weber Problem.** In the Weber (or Fermat-Weber) problem, the coordinates of a number of *facilities* z_i, $i = 1, \ldots, n$ in $\mathbb{R}^p$ are known and the goal is to locate a *new facility* x, so that

$$\mathbf{pull}_1(x) = \sum_{i=1}^n w_i d(x, z_i) \tag{7}$$

is minimized, where $d(x, z_i)$ denotes the distance between the location of the new facility x and an existing one z_i. It is not necessary that the distance d be Euclidean. In fact, a great deal of attention is given to non-Euclidean distances such as the city block or ℓ_1 metric [Francis et al., 1992].

It is also not necessarily true that this *minisum* formulation is the most natural one. In some cases *minimax* is a more direct translation of what we

want to obtain. In that case

$$\overline{\mathbf{pull}}_1(x) = \max_{i=1}^{n} w_i d(x, z_i) \tag{8}$$

must be minimized. It can be seen that the Weber problem corresponds to drawing a weighted star graph using a Tutte normalization.

4.2. **Multifacility Weber Problems.** An obvious generalization is to locate more than one facility (server) in the space, but this leads to various complications, because we do not only want clients and servers to be close, we may also want servers to be relatively far apart. For instance, if we are locating m servers, we may want to minimize

$$\underline{\mathbf{pull}}_1(X) = \sum_{i=1}^{n} w_i \min_{j=1}^{m} d(x_j, z_i), \tag{9}$$

where X contains the coordinates of the m servers in $\mathbb{R}^p$. This minimizes the sum of the distances of the clients to the closest server. If we are locating toilets in a campground, for instance, this seems to be the appropriate criterion. In other cases it may make sense to minimize

$$\overline{\mathbf{pull}}_1(X) = \sum_{i=1}^{n} w_i \max_{j=1}^{m} d(x_j, z_i), \tag{10}$$

for instance if each client must always visit all servers. There is alternative way to define the minimin and minimax loss functions. Let

$$\mathbf{pull}_1(X, W) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} d(x_j, z_i). \tag{11}$$

Then

$$\underline{\mathbf{pull}}_1(X) = \min\{\mathbf{pull}_1(X, W) | \sum_{j=1}^{m} w_{ij} = w_i, w_{ij} \geq 0\}, \tag{12a}$$

$$\overline{\mathbf{pull}}^1(X) = \max\{\mathbf{pull}_1(X, W) | \sum_{j=1}^{m} w_{ij} = w_i, w_{ij} \geq 0\}. \tag{12b}$$

In Francis et al. [1992, Chapter 6] the multifacility location problem is defined as

$$\mathbf{pull}^1(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} v_{ij} d(x_i, x_j) + \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} d(x_j, z_i) \tag{13}$$

The multifacility location problem corresponds to drawing the graph of a weighted complete bipartite graph using the Tutte normalization.

4.3. **Reciprocal Location.** Heiser [1981, 1987b] introduces the *reciprocal location problem* as one way to discuss generalizations of correspondence analysis and multidimensional unfolding. In the reciprocal location problem we have a bipartite graph, and two configurations X and Y. We minimize

$$\textbf{pull}_\phi(X, Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}\phi(d_{ij}(X, Y)), \tag{14}$$

over X and Y, where $d_{ij}(X, Y)$ is the distance between x_i and y_j.

4.4. **The QA/TSP Problem.** Heiser also discusses in this context the *quadratic assignment problem* (QA). In this case *all* points are in fixed locations $z_i,\ i = 1, \ldots, n$. The following loss function is used

$$\textbf{pull}_2(\Pi, Z) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}d_{\pi(i)\pi(j)}(Z), \tag{15}$$

where π is a permutation vector, i.e. a vector whose components form a permutation of $\{1, 2, \ldots, n\}$. The usual interpretation of the QA problem in location theory is the following: there are n facilities and n locations, and the w_{ij}'s represent the flow of materials from facility i to facility j, while the d_{kl}'s represent the distance from location k to location l. The objective is to find an assignment of all facilities to all locations (i.e. a permutation vector π), so that the total cost of the assignment is minimized. The problem was introduced by Koopmans and Beckmann [1957] and was shown by Sahni and Gonzalez [1976] that it is NP-complete, which implies that finding a polynomial time algorithm to solve it is unlikely. Moreover, other well known NP-complete problems can be formulated as special cases of the QA problem; e.g., the traveling salesman problem (TSP) (where the flow matrix corresponds to the adjacency matrix of a cycle of length n and the distance matrix to that of the TSP), the graph partitioning problem (GP) (where the flow matrix corresponds to the adjacency matrix of two disjoint complete graphs of size $n/2$ and the distance matrix to the adjacency matrix of the GP) [Pardalos et al., 1999]. The QA problem can be cast as a graph drawing problem, with the $2n$ vertices having fixed positions in $\mathbb{R}^p$ and the goal being to draw the edges between the two sets of vertices so as to minimize the loss function given in (15).

5. Multivariate Descriptive Statistical Analysis

5.1. **Correspondence Analysis.**

5.2. **Multidimensional Scaling.** In metric (least squares) multidimensional scaling [Borg and Groenen, 1997, de Leeuw and Heiser, 1980] the problem is to minimize the following function

$$\textbf{pull}_\rho(Z) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\rho(d_{ij}(Z)), \tag{16a}$$

where

$$\rho(d_{ij}(Z)) = (\eta(\delta_{ij}) - \eta(d_{ij}(Z)))^2, \tag{16b}$$

Here the δ_{ij} are given numbers (known as *dissimilarities*). They could, for instance, be the path-length distances in the graph. The purpose of the technique is to approximate a matrix of (transformed) dissimilarities by a matrix of (transformed) Euclidean distances in low-dimensional space. Then transformation η is usually the identity, the square, or the logarithm.

Obviously $\rho(d)$ is not increasing and does not pass through zero. Thus it is different from the **pull** functions we have seen so far. Some further analysis is required. By expanding the square we see that minimizing $\textbf{pull}_\rho(Z)$ is equivalent to minimizing $\textbf{pullpush}_{\phi\psi}(Z)$ with $\phi(d) = \eta^2(d)$, $\psi(d) = \eta(d)$, and $u_{ij} = 2\eta(\delta_{ij})w_{ij}$. Thus all points are pulling together, but points with large dissimilarities are being pushed apart.

5.3. **Cluster Analysis.** Location problems, especially the multisource Weber problems corresponds with forms of cluster analysis [Taillard, 1996]. Suppose we have n objects with coordinates x_i, $i = 1, \ldots, n$ in $\mathbb{R}^p$ that we want to group in K clusters. Let C be a $K \times p$ matrix containing the centers of the K clusters in $\mathbb{R}^p$. Then we want to minimize the following function with respect to C,

$$\textbf{clus} = \sum_{i=1}^{n} \min_{k=1}^{K} w_i d(x_i, c_j), \tag{17}$$

where w_i are a set of weights. If d is the square of the distance then we recover the well known sum of squares clustering problem [MacQueen, 1967].

5.4. **Regression Analysis.** Formally, we can also distinguish the case in which the graph is bipartite, and there is a one-one correspondence between its two components [Heiser, 1987b, Verboon, 1990]. We also suppose vertices are connected if and only if they are in correspondence. Under these circumstances, we can drop the double indexing, and we find

$$\textbf{pull}_\phi(X, Y) = \sum_{i=1}^{n} w_i\phi(\|x_i - y_i\|). \tag{18}$$

If we now assume, in addition, that we work in $\mathbb{R}^1$, that the x_i are fixed and known, and that the y_i are constrained by $y_i = z_i'\beta$ or more generally by $y_i = \psi(z_i, \beta)$, then we are in the regression situation. The construction is a bit artificial, because of the many additional assumptions, but actually minimizing the length of the edges in the graph is the same as minimizing the residuals in the regression analysis. The majorization algorithms in this case become iterative reweighted least squares algorithms, corresponding in most cases to the ones discussed, for example, in Holland and Welsch [1977].

A more general discussion of loss function (18) is Verboon [1994]. He extends the majorization approach to $\mathbb{R}^p$, which makes it possible to cover target rotation and canonical analysis techniques.

6. Using Squared Euclidean Distance

Let us first study the case in which $\phi(d_{ij}) = \frac{1}{2}d_{ij}^2$. This is, as we shall see further on, the most important case from the algorithmic point of view, and many more general cases will be reduced to this one. The following notation is convenient: let $d_{ij}^2(Z) = \mathbf{tr}(Z'A_{ij}Z)$ with $A_{ij} = (e_i - e_j)(e_i - e_j)'$ and where the e_i's are unit vectors.

Define the matrix O as

$$O = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}A_{ij}. \tag{19}$$

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

$$\mathbf{pull}_2(Z) = \mathbf{tr}(Z'OZ). \tag{20}$$

This must be minimized under some normalization condition on Z.

The results in the previous section suggest that using the normalization condition $\mathbf{tr}(Z'Z) = 1$ is natural. Unfortunately, this does not work. For $\mathbf{pull}_2$, for instance, we find the stationary equations $OZ = \lambda Z$, which implies that all columns of Z are proportional to the eigenvector corresponding to the smallest non-zero eigenvalue of V. Thus the optimal Z is of rank one. In order to prevent this from happening, we can choose other scalar normalizations such as $\mathbf{det}(Z'Z) = 1$. Or we can choose $Z'Z = I$. These basically all result in Z being equal to the p eigenvectors corresponding to the p smallest nonzero eigenvalues of V.

6.1. The Laplacian Connection.

Some algebra shows that $O = \frac{1}{2}L$, where $L = D - W$, with D being a diagonal matrix containing the degrees of the vertices in V. Define $\mathcal{L} = T^{-1/2}LT^{-1/2}$. This is the Laplacian of the graph

G, an object of intense study over the last 20 years (starting with Fiedler in 1973). Notice that $\mathbf{tr}\mathcal{L} = n$. In the literature the vectors $Z(:, i)$ are also known as Fiedler vectors. We discuss next some properties of $\mathcal{L}$.

1. $\lambda_1 = 0$ withe the corresponding eigenvector $T^{-1/2}u$ with u comprised of all ones.
2. $\lambda_2 > 0$ iff the graph is connected.
3. $\lambda_n = 2$ iff the graph is bipartite.
4. For the complete graph K_n, the eigenvalues are 0, and $n/(n-1)$ with multiplicity $n-1$.
5. For the complete bipartite graph K_{n_1,n_2}, the eigenvalues are 0, 1 with multiplicity $n_1 + n_2 - 2$ and 2.
6. For the star graph on n vertices, the eigenvalues are 0, 1 with multiplicity $n-2$ and 2.
7. For the n dimensional hypercube on 2^n vertices, the eigenvalues are $2k/n$, with multiplicity $\binom{n}{k}$ for $k = 0, 1, \ldots, n$

In general, the second eigenvalue λ_2 provides a lot of information for the underlying graph. The larger its value is the more connected the components of the graph are and therefore the harder to split it; thus implying that clustering a dataset with a large λ_2 is hard. However, the converse is not true, since a highly connected graph with a single isolated vertex will necessarily have $\lambda_2 = 0$. Moreover, the eigenvalues and eigenvectors of the Laplacian have been succesfully used in isoperimetric problems, path, flow and routing problems, construction of graph exapnders [Chung, 1997], in seriation problems [de Leeuw and Michailidis, 1999], etc.

6.2. **Partitioned Normalization.** Suppose Z is partitioned as

$$Z = \begin{bmatrix} X \\ Y \end{bmatrix} \tag{21}$$

where X is normalized in some way, and Y is free. Then, O can be partitioned as follows

$$O = \begin{bmatrix} O_{11} & O_{12} \\ O_{12}' & O_{22} \end{bmatrix}, \tag{22}$$

and the pull function can be written as

$$\mathbf{pull}_2(X, Y) = \mathbf{tr}(X'O_{11}X) + 2\mathbf{tr}(X'O_{12}Y) + \mathbf{tr}(Y'O_{22}Y), \tag{23}$$

and thus

$$\mathbf{pull}_2(X, \star) = \min_Y \sigma(X, Y) = \mathbf{tr}(X'\{O_{11} - O_{12}'O_{22}^{-1}O_{12}\}X). \tag{24}$$

and this quadratic must still be minimized over normalized X.

An example is the normalization proposed by Tutte [1963]. In the *Tutte Normalization*, X is simply fixed at some value (provided that X contains

at least 3 points), and the optimal Y is chosen by solving the above problem. Clearly if the normalization actually fixes X, there is no need to minimize (24). However, the choice of which points to fix can be somewhat arbitrary, and it has a great deal of influence on the solution [de Leeuw and Michailidis, 1999, Healy and Goldstein, 1976].

Remark . Under the Tutte normalization (X fixed), from (23) we get that

$$\nabla \mathbf{pull}_2(Y) = O_{22}Y + O_{12}'X = 0, \tag{25}$$

which can be rewritten as

$$O_{22}Y = -O_{12}'X. \tag{26}$$

The latter implies that we need to solve systems of linear equations of the form $O_{22}Y(:,s) = -O_{12}'X(:,s),\ s = 1,\ldots,m$. Since O_{22} is strictly diagonally dominant (provided we are dealing with a graph without any isolated vertices) and symmetric, it follows that it is positive definite. Moreover, in practice it can be very large and usually extremely sparse; hence, the preferred methods for solving such systems are iterative solvers based on Krylov subspace methods Barrett et al. [1994]. For the specific problem at hand, the nature of the O_{22} matrix gurantees that so-called "direct" iterative methods such as Jacobi and Gauss-Seidel will perform well.

On the other hand, under the normalization $X'X = I_m$, from (24) we get that we have to compute the first eigenvalues and eigenvectors of a possibly fairly large and sparse matrix and Lanczos based methods come in handy Golub and Loan [1997].

6.3. **Bipartite Graphs.** For bipartite graphs the W matrix is given by

$$W = \begin{bmatrix} 0 & A \\ A' & 0 \end{bmatrix}, \tag{27}$$

where A denotes the adjacency matrix of the graph. We then have that

$$O = \begin{bmatrix} D_r & -A \\ -A' & D_c \end{bmatrix}, \tag{28}$$

with D_r and D_c diagonal matrices containing the row sums and column sums of A respectively. In case the bipartite graph corresponds to the classical data structure discussed in Section 1.4, we get that A corresponds to the *superindicator* matrix [Gifi, 1990] and $D_r = JI_N$, where I_N denotes the identity matrix of order N, while A is the contingency table itself for bivariate datasets. Then, the pull function becomes

$$\mathbf{pull}_2(X,Y) = \mathbf{tr}(X'DX) - 2\mathbf{tr}(X'AY) + \mathbf{tr}(Y'EY). \tag{29}$$

Thus the minimum over X for fixed Y is given by

$$X = D_r^{-1}AY, \tag{30a}$$

while the minimum over Y for fixed X is given by

$$Y = D_c^{-1} A' X. \tag{30b}$$

These are the two *centroid principles* extensively discussed in de Leeuw et al. [1999]. They are familiar from discussion of correspondence analysis, but we see that they apply much more generally. Therefore, partitioned normalization forces the free points to be located in the *convex hull* of the normalized ones, a very desirable feature from a drawing point of view (see examples in Michailidis and de Leeuw [1999a]).

7. Euclidean Distance without the Square

A particularly interesting pull function is

$$\mathbf{pull}_1(Z) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} d_{ij}(Z). \tag{31}$$

This minimization problem can be easily solved by using majorization based on the AM/GM inequality.

$$d_{ij}(Z) \leq \frac{1}{2} \frac{1}{d_{ij}(Y)} (d_{ij}^2(Z) + d_{ij}^2(Y)), \tag{32}$$

and thus

$$\mathbf{pull}_1(Z) \leq \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{w_{ij}}{d_{ij}(Y)} (d_{ij}^2(Z) + d_{ij}^2(Y)) = \frac{1}{2} \{\mathbf{tr}\ Z' B(Y) Z + \mathbf{tr}\ Y' B(Y) Y\},$$

where

$$B(Y) = \sum_{i=1}^n \sum_{j=1}^n \frac{w_{ij}}{d_{ij}(Y)} A_{ij}. \tag{33}$$

Thus in an iteration we minimize $\mathbf{tr}\ Z' B(Z^{previous}) Z$ over normalized Z.

In 1937 Weiszfeld proposed this algorithm for solving the Weber problem with Euclidean distances. Eckhardt [Eckhardt, 1980] established the global linear convergence of this algorithm in general Banach spaces, and Voss and Eckhardt [1980] generalized the algorithm to the multifacility Weber problem.

Heiser [1986] (independently, it seems) proposed the same algorithm in the data analysis context of the reciprocal location problem. He also discussed [Heiser, 1987a] the problems with zero distances, and proposed a solution similar to, but slightly less straightforward, than the classical hyperbolic perturbation.

7.1. **Improving Convergence Speed.** The majorization algorithms introduced in this paper have a linear rate of convergence, which for a generic scalar convergent sequence $\{z_n\}$ implies that

$$\lim_{k\rightarrow\infty}\frac{z_{k+1}-z_\infty}{z_k-z_\infty}=\lambda,\ \text{for some } |\lambda|\in(0,1). \tag{34}$$

However, in large problems such a rate of convergence is prohibitively slow. We discuss next strategies to increase the convergence speed of linearly convergent sequences. One of the oldest schemes is the Δ^2 transformation of Aitken [Delahaye, 1988] given by

$$t_k=\frac{z_{k+2}z_k-z_{k+1}^2}{z_{k+2}-2z_{k+1}+z_k}, \tag{35}$$

provided that $z_{k+2}-2z_{k+1}+z_k\neq 0$. In Delahaye [1988], it is shown that for linearly convergent sequences Aitken's transformation is optimal.

7.2. **Alternative Algorithms.**

7.3. **Logarithm of Distance.** Suppose

$$\mathbf{pull}_{log}(Z)=\sum_{i=1}^n\sum_{j=1}^n w_{ij}\log d_{ij}(Z)$$

Again we minimize this by using majorization. A first possibility is to use

$$\log d_{ij}(Z)\leq\log d_{ij}(Y)+\frac{1}{d_{ij}(Y)}(d_{ij}(Z)-d_{ij}(Y)).$$

This implies that we iteratively minimize

$$\mathbf{pull}_1(Z)=\sum_{i=1}^n\sum_{j=1}^n\frac{w_{ij}}{d_{ij}(Y)}d_{ij}(Z).$$

by the methods explained for $\mathbf{pull}_1$.

It may be more convenient to use

$$\log d_{ij}^2(Z)\leq\log d_{ij}^2(Y)+\frac{1}{d_{ij}^2(Y)}(d_{ij}^2(Z)-d_{ij}^2(Y)),$$

which we can also write as

$$\log d_{ij}(Z)\leq\log d_{ij}(Y)+\frac{1}{2d_{ij}^2(Y)}(d_{ij}^2(Z)-d_{ij}^2(Y)).$$

This amounts to minimizing

$$\mathbf{pull}_2(Z)=\sum_{i=1}^n\sum_{j=1}^n\frac{w_{ij}}{d_{ij}^2(Y)}d_{ij}^2(Z),$$

in each iteration and this is a quadratic problem of the form **tr** $Z'H(Y)Z$, where

$$H(Y) = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{w_{ij}}{d_{ij}^2(Y)} A_{ij}.$$

Observe that using this second majorization gives a less precise approximation than the first, and consequently may lead to slower convergence.

7.4. **Weiszfeld's Algorithm for the Weber Problem.** In 1937 Weiszfeld proposed the following interative algorithm for solving the Weber problem with Euclidean distances (i.e. $d(x, z_i) = \|x - z_i\|$).

Step 0: Pick some initial location $x^{(0)} \in \mathbb{R}^p$.

Step 1: At step k, find x^* that minimizes

$$\psi(x, x^{(k)}) = \sum_{i=1}^{n} w_i \frac{\|x - z_i\|^2}{\|x^{(k)} - z_i\|}. \tag{36}$$

Step 2: Set $x^{(k+1)} = x^*$ and go back to step 1, until convergence.

Eckhardt [Eckhardt, 1980] established the global linear convergence of this algorithm in general Banach spaces, and Voss and Eckhardt [1980] generalized the algorithm to the multifacility Weber problem.

Remark . We present next a Newton primal-dual algorithm for the Weber multifacility problem under the $\mathbf{pull}_1$ loss function.

8. Majorization Methods

8.1. **General Principles.** The algorithms proposed in this paper are all of the majorization type. Majorization is discussed in general terms in de Leeuw [1994], Heiser [1995], Lange et al. [2000].

In a majorization algorithm we want to optimize a function $\phi(\theta)$ over $\theta \in \Theta$, with $\Theta \subseteq \mathbb{R}^p$. Suppose that a function $\psi(\theta, \xi)$ defined on $\Theta \times \Theta$ satisfies

$$\phi(\theta) \geq \psi(\theta, \xi) \text{ for all } \theta, \xi \in \Theta, \tag{37a}$$

$$\phi(\theta) = \psi(\theta, \theta) \text{ for all } \theta \in \Theta. \tag{37b}$$

Thus, for a fixed ξ, $\psi(\bullet, \xi)$ is below ϕ, and it touches ϕ at the point $(\xi, \phi(\xi))$. We then say that $\phi(\theta)$ *majorizes* $\psi(\theta, \xi)$ or that $\psi(\theta, \xi)$ *minorizes* $\phi(\theta)$.

There are two key theorems associated with these definitions.

Theorem 8.1. *If ϕ attains its maximum on Θ at $\hat{\theta}$, then $\psi(\bullet, \hat{\theta})$ also attains its maximum on Θ at $\hat{\theta}$.*

Proof. Suppose $\psi(\tilde{\theta},\hat{\theta}) > \psi(\hat{\theta},\hat{\theta})$ for some $\tilde{\theta} \in \Theta$. Then, by (37a) and (37b), $\phi(\tilde{\theta}) \geq \psi(\tilde{\theta},\hat{\theta}) > \psi(\hat{\theta},\hat{\theta}) = \phi(\hat{\theta})$, which contradicts the definition of $\hat{\theta}$ as the maximizer of ϕ on Θ. □

Theorem 8.2. *If $\tilde{\theta} \in \Theta$ and $\hat{\theta}$ maximizes $\psi(\bullet,\tilde{\theta})$ over Θ, then $\phi(\hat{\theta}) \geq \phi(\tilde{\theta})$.*

Proof. By (37a) we have $\phi(\hat{\theta}) \geq \psi(\hat{\theta},\tilde{\theta})$. By the definition of $\hat{\theta}$ we have $\psi(\hat{\theta},\tilde{\theta}) \geq \psi(\tilde{\theta},\tilde{\theta})$. And by (37b) we have $\psi(\tilde{\theta},\tilde{\theta}) = \phi(\tilde{\theta})$. Combining these three results we get the result. □

These two results suggest the following algorithm for maximizing $\phi(\theta)$.

Step 1:: Given a value $\theta^{(k)}$ construct a minorizing function $\psi(\theta^{(k)},\xi)$.
Step 2:: Maximize $\psi(\theta^{(k)},\xi)$ with respect to ξ. Set $\theta^{(k+1)} = \xi^{\max}$.
Step 3:: If $|\phi(\theta^{(k+1)} - \phi(\theta^{(k)}| < \epsilon$ for some predetermined $\epsilon > 0$ stop; else go to Step 1.

In order for this algorithm to be of practical use, the minozrizing function ψ needs to be easy to maximize, otherwise nothing substantial is gained by following this route. Notice, that in case we are interested to minimize ϕ, we have to find a majorizing function ψ that needs to be minimized in Step 2.

We demonstrate next how the idea behind majorization works with a simple example.

Example . This is an artificial example, chosen for its simplicity. Consider $\phi(\theta) = \theta^4 - 10\theta^2,\ \theta \in \mathbb{R}$. Because $\theta^2 \geq \xi^2 + 2\xi(\theta - \xi) = 2\xi\theta - \xi^2$ we see that $\psi(\theta,\xi) = \theta^4 - 20\xi\theta + 10\xi^2$ is a suitable majorization function. The majorization algorithm is $\theta^+ = \sqrt[3]{5\xi}$.

The algorithm is illustrated in Figure 8.1. We start with $\theta(0) = 5$. Then $\psi(\theta,5)$ is the dashed function. It is minimized at $\theta^{(1)} \approx 2.924$, where $\psi(\theta^{(1)},5) \approx 30.70$, and $\phi(\theta^{(1)}) \approx -12.56$. We then majorize by using the dotted function $\psi(\theta,\theta^{(1)})$, which has its minimum at about 2.44, equal to about -21.79. The corresponding value of ϕ at this point is about -24.1. Thus we are rapidly getting close to the local minimum at $\sqrt{5}$, with value 25. The linear convergence rate at this point is $\frac{1}{3}$.

We briefly address next some convergence issues (for a general discussion see the book by Zangwill [1969] and also Meyer [1976]). If ϕ is bounded above (below) on Θ, then the algorithm generates a bounded increasing sequence of function values $\phi(\theta^{(k)}$, thus it converges to $\phi(\theta^{\infty})$. For example, continuity of ϕ and compactness of Θ would suffice for establishing the result. Moreover with some addiitonal mild continuity considerations we get that $||\theta^{(k)} - \theta^{(k+1)}|| \to 0$ de Leeuw [1990], which in turn implies, due to a result by Ostrowski Ostrowski [1966], that θ converges either to a stable point or to a continuum of limit points. Hence, majorization algorithms

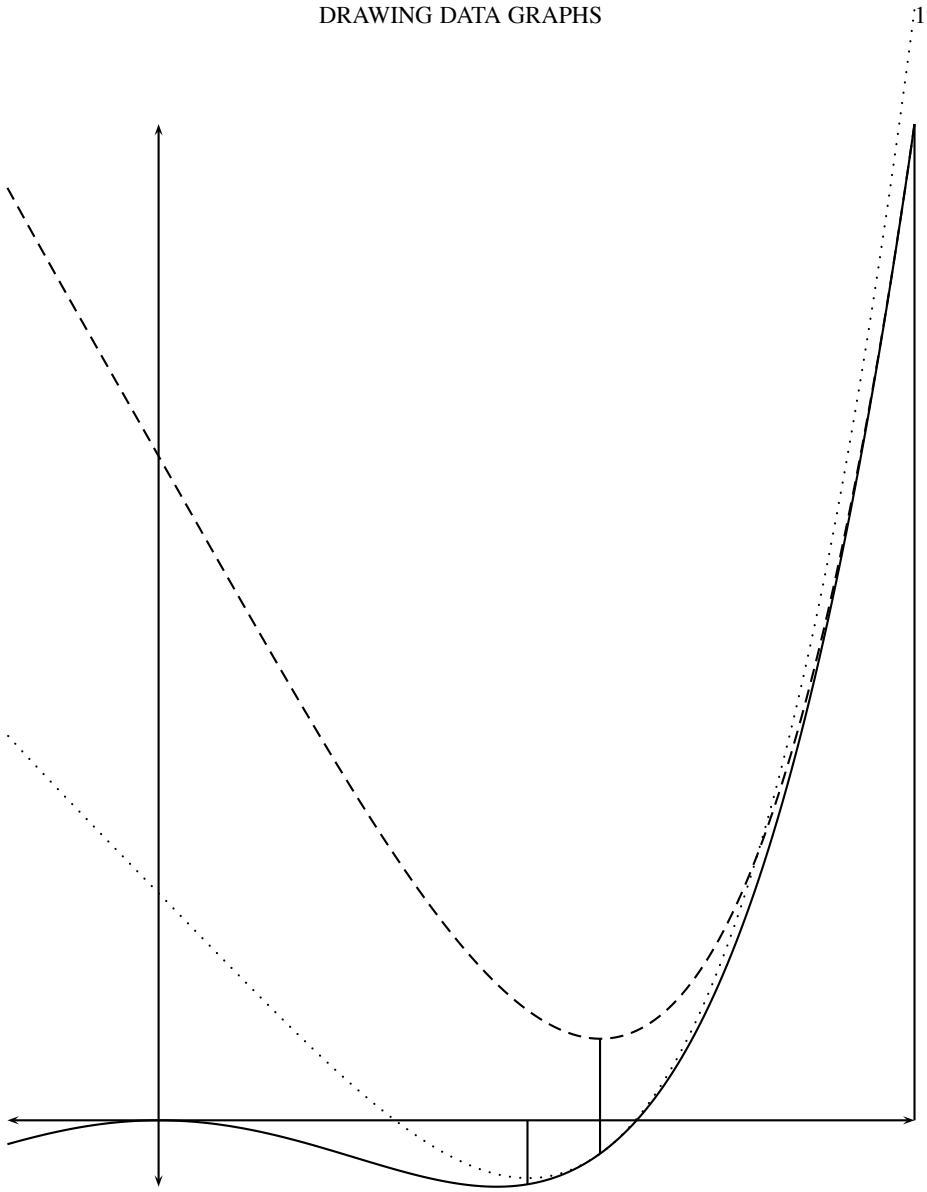

FIGURE 1. Majorization

for all practical purposes find local optima, and by starting the algorithm at different initial values global optima can be located.

We turn next our attention to issues regarding rates of convergence.

Theorem 8.3. *This implies that* $\mathcal{D}_2(\omega, \omega) = 0$ *for all* ω, *and consequently* $\mathcal{D}_{12} = -\mathcal{D}_{22}$. *Thus* $\mathcal{M} = -\mathcal{D}_{11}^{-1}\mathcal{D}12$.

8.2. **Using Convexity.** Suppose $\phi(\theta)$ is a convex function. We then have for ϕ finite in ξ that

$$\phi(\theta) \geq \phi(\xi) + < \mathcal{D}, \theta - \xi >, \tag{38}$$

$\mathcal{D}$ is the subgradient of ϕ at ξ. The *subgradient inequality* Rockafellar [1970] says that the graph of the *affine* function $h(\xi) = \phi(\theta) + < \mathcal{D}, \theta - \xi >$ is a non-vertical supporting hyperplane to the convex set of the epigraph of ϕ at the point $(\xi, \phi(\xi))$. The set of all subgradients of ϕ at ξ is called the *subdifferential* of ϕ at ξ and is denoted by $\partial\phi(\xi)$. Obviously $\partial\phi(\xi)$ is a closed convex set, since by definition $\mathcal{D} \in \partial\phi(\xi)$ if and only if $\mathcal{D}$ satisfies a certain infinite system of weak linear inequalities (one for each θ). In general $\partial\phi(\xi)$ may be empty or it may consist of just one vector. Similarly for concave functions we have that

$$\phi(\theta) \leq \phi(\xi) + < \mathcal{D}, \theta - \xi >, \tag{39}$$

with $\mathcal{D} \in \partial\phi(\xi)$. Hence, concave functions have a linear majorizing function $\psi(\theta, \xi) = \mathcal{D}\theta$.

Another important class of functions are the d.c. (difference of convex functions) ones (see Appendix B), defined by $\phi(\theta) = g(\theta) - h(\theta)$, with g, h convex functions. We can then write

$$\phi(\theta) \leq g(\theta) - h(\xi) - < \mathcal{D}, \theta - \xi >, \tag{40}$$

with $\mathcal{D} \in \partial h(\xi)$. This provides a convex majorizer $\psi(\theta, \xi) = g(\theta) - < \mathcal{D}, \theta >$.

8.3. **Strongly Convex Functions.** This class of functions satisfies the following inequality Hiriart-Urruty and Lemarechal [1993]

$$\phi(\theta) \geq \phi(\xi) + < \nabla\phi(\xi), \theta - \xi > + \frac{1}{2} M ||\theta - \xi||^2, \tag{41}$$

with modulus M on Θ. Functions with bounded second derivatives belong to this class (i.e. $\nabla^2\phi(\theta) < M$). For strongly concave functions that are of interest to us we similarly get

$$\phi(\theta) \leq \phi(\xi) + < \nabla\phi(\xi), \theta - \xi > + \frac{1}{2} M ||\theta - \xi||^2, \tag{42}$$

which after defining after defining $\eta(\xi) = \theta - M^{-1}\nabla\phi(\xi)$ can be written as

$$\phi(\theta) \leq \phi(\xi) - \frac{1}{2} M^{-1}\nabla\phi(\xi) + \frac{1}{2} ||\theta - \eta(\xi)||^2, \tag{43}$$

which shows that we can define a quadratic majorizing function $\psi(\theta, \xi) = ||\theta - \eta(\xi)||^2$.

8.4. **Convex Functions with Slow Growth Rates.**

Lemma 8.4. *Let $h : \mathbb{R}_+ \to \mathbb{R}_+$ be continuous and strictly increasing with $h(0) = 0$, let k be the inverse of h, and define $H(x) = \int_0^x h(y)dy$ and $K(x) = \int_0^x k(y)dy$. Then, for all $a, b \in \mathbb{R}_+$, $ab \leq H(a) + K(b)$, with equality if and only if $b = h(a)$.*

Proof. Suppose $b \leq h(a)$. Let $c = h^{-1}(b)$; therefore, $c < a$. Then

(44)
$$H(a) = \int_0^a h(x)dx = \int_0^c h(x)dx + \int_c^a h(x)dx \geq \int_0^c h(x)dx + b(a - c).$$

The inequality in (44) is strict, unless $a = c$. Also, by the change of variables $x = k(y)$, we get

$$K(b) = \int_0^b k(y)dy = \int_0^c xh'(x)dx. \tag{45}$$

But,

$$\int_0^c xh'(x)dx + \int_0^c h(x)dx = \int_0^{cb} du = cb, \tag{46}$$

by the change of variables $u = xh(x)$. Combining (44) and (45) we get that $H(a) + K(b) \geq cb + b(a - c) = ab$.

However, a geometric proof is immediate (see Figure below), if we interpret each term as an area and remember that the graph of h also serves as that of k if we interchange the x and y axes. Equality holds if and only if the point (a, b) lies on the graph of h.

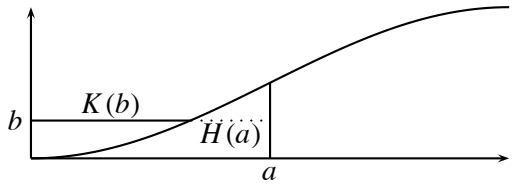

FIGURE 2. Geometric proof of Young's Inequality

□

Remark . (AM-GM Inequality:) If $g(x) = \sqrt{x}$, then we get

$$\sqrt{ab} \leq \frac{1}{2}(a + b), \tag{47}$$

and hence recover the so-called Arithmetic Mean - Geometric Mean inequality.

Remark . *(Holder Inequality:)* Suppose that $p, q > 1$ such that

$$\frac{1}{p} + \frac{1}{q} = 1. \tag{48}$$

Then, for all $a, b > 0$ we have

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}, \tag{49}$$

with equality if and only if $b = a^{p-1}$. This follows by considering $g(x)x^{p-1}$. For $p = q = 2$, we recover the Cauchy-Schwarz inequality.

8.5. **Some Useful Results.** Suppose $\Theta_1, \ldots, \Theta_m$ are disjoint subsets of $\mathbb{R}^p$, and $\phi_i : \Theta_i \Rightarrow \mathbb{R}$ are real valued functions. Define

$$\phi(\theta) = \sum_{i=1}^{m} \delta_i(\theta) f_i(\theta), \tag{50}$$

where $\delta_i(\theta) = 1$ if $\theta \in \Theta_i$ and 0 otheriwse.

Theorem 8.5. *Suppose $\psi_i(\theta, \xi)$ majorizes $\phi_i(\theta)$ on Θ_i, i.e.*

$$\phi_i(\theta) \leq \psi_i(\theta, \xi) \text{ for all } \theta \in \Theta_i \text{ and all } \xi \in \mathbb{R}^p, \tag{51}$$

$$\phi_i(\theta) = \psi_i(\theta, \theta) \text{ for all } \theta \in \Theta_i. \tag{52}$$

Then,

$$\psi(\theta, \xi) = \sum_{i=1}^{m} \delta_i(\theta)\psi_i(\theta, \xi) \tag{53}$$

majorizes $\phi(\theta)$ on $\Theta = \cup_{j=1}^{m}\Theta_i$.

Proof. Suppose $\theta \in \Theta_i$. Then $\phi(\theta) = \phi_i(\theta) \leq \psi_i(\theta, \xi) = \psi(\theta, \xi)$ for all $\xi \in \mathbb{R}^p$. Also $\phi(\theta) = \phi_i(\theta) = \psi_i(\theta, \theta) = \psi(\theta, \theta)$. □

Theorem 8.6. *Suppose $\phi(x)$ is an increasing concave function. Let $\psi(x) = \phi(\sqrt{x})$. Then $\psi(x)$ is also concave and increasing.*

Proof. By defintion $\psi(\lambda x + (1-\lambda)) = \phi(\sqrt{\lambda x + (1-\lambda)y})$. But the square root is concave, and thus $\sqrt{\lambda x + (1-\lambda)y} \geq \lambda\sqrt{x} + (1-\lambda)\sqrt{y}$, and because ϕ is increasing $\psi(\lambda x + (1-\lambda)y) \geq \phi(\lambda\sqrt{x} + (1-\lambda)\sqrt{y})$. Finally, because ϕ is concave, $\psi(\lambda x + (1-\lambda)y) \geq \lambda\psi(x) + (1-\lambda)\psi(y)$. □

Remark . The above Theorem implies that if ϕ is a squasher (see Section 9.2), and thus

$$\mathbf{pull}_\phi(Z) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\phi(d_{ij}^2(Z))$$

is concave with a linear majorizer, then ψ is a squasher too, while

$$\mathbf{pull}_\psi(Z) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}\psi(d_{ij}^2(Z)) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}\phi(d_{ij}(Z))$$

is concave too with a linear majorizer.

Suppose f_s are convex, with $s = 1, \ldots, p$. Define

$$g_0 = \sum_{s=1}^p f_s,$$
$$g_r = \sum_{s=1}^p f_s - f_r,$$

and also

$$h_0 = \min_{s=1}^p f_s,$$
$$h_1 = \max_{s=1}^p g_s.$$

Clearly, all off $g_0, \ldots, g_p$ are convex, and so is h_1. Also, trivially,

$$h_0 = g_0 - h_1.$$

This represents the minimum of the f_s as a d.c. (difference of convex functions) function. Thus we can majorize h_0 by a convex function, using

$$h_0(x) \leq g_0(x) - h_1(y) - \mathcal{D}h_1(y)(x - y) = g_0(x) - h_1(y) - \mathcal{D}g_{r(y)}(y)(x - y),$$

where $r(y)$ is such that

$$g_{r(y)}(y) = h_1(y).$$

This result can be applied in a straightforward way to the multifacility location problem, which is minimization of

$$\sigma(x_1, \ldots, x_p) = \sum_{i=1}^n w_i \min_{s=1}^p \|z_i - x_s\|,$$

where the z_i are existing facilities and the x_s are new facilities (to be located).

9. Constructing Pull Majorizing Functions

In order to minimize the various **pull** functions we employ the majorization principle, discussed in the Appendix. Finding a majorizing function for an arbitrary function is, to a certain extent, and art. However, in the Appendix we present some systematic ways to find majorizing functions for the pull function. We now apply them to various **pull** functions.

9.1. **A interesting class of functions.** A particularly interesting class of pull functions is given by

$$\mathbf{pull}_\beta(Z) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} d_{ij}^\beta(Z),\ \beta \in [1,2], \tag{54}$$

These are convex functions with growth rates slower than the quadratic. The class contains as extreme cases both the $\mathbf{pull}_2$ function and the $\mathbf{pull}_1$ function that deals with Euclidean distances (distances without the square). This minimization problem can be easily solved by using majorization based on Young's inequality (see Section 8.4 in the Appendix).

$$d_{ij}^\beta(Z) \leq \frac{2-\beta}{2} d_{ij}^\beta(Y) + \frac{2}{\beta d_{ij}^{2-\beta}(Y)} d_{ij}^2(Z), \tag{55}$$

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

$$\mathbf{pull}_\beta(Z) \leq \sum_{i=1}^n \sum_{j=1}^n w_{ij} \big(\frac{2-\beta}{2} d_{ij}^\beta(Y) + \frac{2}{\beta d_{ij}^{2-\beta}(Y)} d_{ij}^2(Z) \big) = \mathbf{tr}\big(\frac{2-\beta}{2} Y' B_\beta(Y) Y + \frac{2}{\beta} Z' B_\beta(Y) Z \big),$$

where

$$B_\beta(Y) = \sum_{i=1}^n \sum_{j=1}^n \frac{w_{ij}}{d_{ij}^{2-\beta}(Y)} A_{ij},\ \beta \in [1,2]. \tag{56}$$

Thus, in an iteration we minimize $\mathbf{tr}(Z' B_\beta(Z^{previous}) Z)$ over normalized Z.

9.2. **Squashers.** Squashers are increasing concave functions passing through the origin. To keep the discussion simple, we also assume that they belong to $\mathcal{C}^2(\mathbb{R}_+)$ (i.e. $\phi(0) = 0,\ \phi'(d) > 0$ and $\phi''(d) < 0,\ \forall d \in \mathbb{R}_+$). Examples of such functions are $\phi(d) = d/(1+d),\ \phi(d) = d^\beta,\ \beta \in (0,1)$, the logistic function $\phi(d) = e^d/(1+e^d)$, etc. We also treat $\phi(d) = \log(d)$ as

a squasher, although it does not pass through the origin. Hence we want to minimize

$$\mathbf{pull}_\phi(Z) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}\phi(d_{ij}(Z)).$$

Using the results outlined in Section 8.2 we get that

$$\phi(d_{ij}(Z)) \leq \phi(d_{ij}(Y)) + \phi'(d_{ij}(Y))(d_{ij}(Z) - d_{ij}(Y)).$$

This implies that we iteratively minimize

$$\mathbf{pull}_1(Z) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}\phi'(d_{ij}(Y))d_{ij}(Z)$$

by the methods explained for $\mathbf{pull}_1$.

It many cases it may be more convenient or appropriate to use squared distances instead. Thus we get

$$\phi(d_{ij}^2(Z)) \leq \log(d_{ij}^2(Y)) + \frac{1}{2}\phi'(d_{ij}^2(Y))(d_{ij}^2(Z) - d_{ij}^2(Y)),$$

which we can also write using the results in 8.6

$$\phi(d_{ij}(Z)) \leq \phi(d_{ij}(Y)) + \frac{1}{2}\phi'(d_{ij}^2(Y))(d_{ij}^2(Z) - d_{ij}^2(Y)).$$

The latter amounts to minimizing

$$\mathbf{pull}_2(Z) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}\phi'(d_{ij}^2(Y))d_{ij}^2(Z),$$

in each iteration and this is a quadratic problem of the form $\mathbf{tr}(Z'H(Y)Z)$, where

$$H(Y) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}(d_{ij}^2(Y))A_{ij}.$$

Observe that using this second majorization gives a less precise approximation than the first, and consequently may lead to slower convergence.

9.3. **Huber and Biweight Functions.** The Huber function is defined as:

$$\phi(d) = \begin{cases} \frac{1}{2}d^2 & \text{if } d < c \\ cd - \frac{1}{2}c^2 & \text{if } d > c \end{cases} \tag{57}$$

for some $c > 0$. The function consists of two parts: for distances smaller than the tuning constant c the squared distance is evaluated, while for large distances the distance itself is used. The basic idea is that outliers, yielding large distances, will have a smaller effect upon the solution and thus a better picture will be obtained. It should be noted that the Huber function

is symmetric around zero in general, but in this case since we deal with distances we only keep its positive part. This is an example of a strongly convex function (having bounded second derivatives). So we have that it is majorized [Verboon, 1994] by

$$\textbf{pull}_\phi(Z) \leq \sum_{i=1}^n \sum_{j=1}^n w_{ij} \frac{1}{2} d(_{ij}^2(Z) \text{ if } d_{ij}(Y) < c$$

$$\textbf{pull}_\phi(Z) \leq \sum_{i=1}^n \sum_{j=1}^n w_{ij} \frac{1}{2} \big[\frac{c}{d_{ij}(Y)} d_{ij}^2(Z) + c d_{ij}(Y) - c^2\big] \text{ if } d_{ij}(Y) > c$$

Another function that limits the influence of outliers on the solution is the biweight function defined as

$$\phi(d) = \begin{cases} \frac{c^2}{6}\big[1 - (1 - (d/c)^2)^3\big] & \text{if } d < c \\ c^2/6 & \text{if } d > c \end{cases} \tag{58}$$

Its first derivative is not monotone, and it becomes zero for distances larger than the tuning parameter c. Verboon [1994] shows that it is majorized by

$$\textbf{pull}_\phi(Z) \leq \sum_{i=1}^n \sum_{j=1}^n w_{ij} \frac{c^2}{6} \big[1 - 3v_{ij}(Y)(1 - (d_{ij}(Z)/c)^2) + 2v_{ij}^{2/3}(Y)\big] \text{ if } d_{ij}(Y) < c$$

$$\textbf{pull}_\phi(Z) \leq \sum_{i=1}^n \sum_{j=1}^n w_{ij} \frac{c^2}{6} \text{ if } d_{ij}(Y) > c$$

where $v_{ij}(Y) = (1 - (d_{ij}/c)^2)^2 2$.

10. Examples

References

R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, 1994.

J.P. Benzécri. *Correspondence analysis handbook*. Marcel Dekker, Inc., New York, New York, 1992.

I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling*. Springer-Verlag, 1997.

F.R.K. Chung. *Spectral Graph Theory*. CBMS Lecture Notes, Providence, Rhode Island, 1997.

J. de Leeuw. Multivariate analysis with optimal scaling. In S. Das Gupta and J. Sethuraman, editors, *Progress in Multivariate Analysis*, Calcutta, India, 1990. Indian Statistical Institute.

J. de Leeuw. Block-relaxation algorithms in statistics. In H.H. Bock, W. Lenski, and M.M. Richter, editors, *Information Systems and Data Analysis*. Springer Verlag, 1994.

J. de Leeuw and W. J. Heiser. Theory of multidimensional scaling. In P.R. Krishnaiah, editor, *Handbook of statistics, volume II*. North Holland Publishing Company, Amsterdam, The Netherlands, 1980.

J. de Leeuw and G. Michailidis. Graph layout techniques and multidimensional data analysis. In T. Bruss and L. LeCam, editors, *Festschrift for Thomas S. Ferguson*. IMS, 1999.

J. de Leeuw, G. Michailidis, and D. Y. Wang. Correspondence analysis techniques. In S. Ghosh, editor, *Multivariate Analysis, Design of Experiments, and Survey Sampling*. Marcel Dekker, 1999.

J.P. Delahaye. *Sequence Transformations*. Springer, Berlin, Germany, 1988.

G. di Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph Drawing; Algorithms for Geometric Representations of Graphs*. Prentice Hall, 1998.

U. Eckhardt. Weber's problem and Weiszfeld's algorithm in general spaces. *Mathematical Programming*, 18:186–196, 1980.

H.A. Eiselt and G. Laporte. Objectives in location problems. In Z. Drezner, editor, *Facility Location. A Survey of Applications and Methods*. Springer, 1995.

R.L. Francis, L.F. McGinnis Jr., and J.A. White. *Facility Layout and Location: An Analytical Approach*. Prentice Hall, 1992. Second Edition.

A. Gifi. *Nonlinear multivariate analysis*. Wiley, Chichester, England, 1990.

G.H. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Univeristy Press (3rd ed), Baltimore, 1997.

M.J.R. Healy and H. Goldstein. An approach to scaling of categorized attributes. *Biometrika*, 63:219–229, 1976.

W. J. Heiser. A majorization algorithm for the reciprocal location problem. Technical Report RR-86-12, Department of Data Theory, University of Leiden, Leiden, The Netherlands, 1986.

W. J. Heiser. Notes on the LARAMP algorithm. Technical Report RR-87-04, Department of Data Theory, University of Leiden, Leiden, The Netherlands, 1987a.

W.J. Heiser. *Unfolding Analysis of Proximity Data*. PhD thesis, University of Leiden, 1981.

W.J. Heiser. Correspondence analysis with least absolute residuals. *Computational Statistica and Data Analysis*, 5:357–356, 1987b.

W.J. Heiser. Convergent computing by iterative majorization: theory and applications in multidimensional data analysis. In W.J. Krzasnowski, editor, *Recent Advances in Descriptive Multivariate Analysis*. Clenrendon Press, Oxford, England, 1995.

J.-B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, New York, 1993.

P.W. Holland and R.E. Welsch. Robust regression using iteratively reweighted least squares. *Communications in Statistics*, A6:813–827, 1977.

T.C. Koopmans and M.J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

K. Lange, D.R. Hunter, and I. Yang. Optimization transfer algorithms in statistics. *Journal of Computational and Graphical Statistics*, pages 00–00, 2000.

J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

R. R. Meyer. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *Journal of Computer and System Sciences*, 12:108–121, 1976.

G. Michailidis and J. de Leeuw. Constrained homogeneity analysis with applications to hierarchical data. Technical Report 207, UCLA, Department of Statistics, Los Angeles, CA, 1997.

G. Michailidis and J. de Leeuw. The Gifi system for descriptive multivariate analysis. *Statistical Science*, 13:307–336, 1999a.

G. Michailidis and J. de Leeuw. Multilevel homogeneity analysis with differential weighting. *Computational Statistics and Data Analysis*, 1999b.

G. Michailidis and J. de Leeuw. A review of graph embeddings. Unpublished Manuscript, Department of Statistics, University of Michigan., 1999c.

A. M. Ostrowski. *Solutions of equations and systems of equations*. Academic Press, New York, New York, 1966.

P.M. Pardalos, F. Rendl, and H. Wolkowicz. The quadratic assignment problem: A survey and recent developments. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 1999.

R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the Association of Computer Machinery*, 23:555–565, 1976.

E.D. Taillard. Heuristic methods for large centroid clustering problem s. Technical Report Technical Report IDSIA 96-96, IDSIA, Lugano, Switzerland, 1996.

W.T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 13:743–767, 1963.

P. Verboon. Majorization with iteratively reweighted least squares: a general approach to optimize a class of resistant loss functions. Technical Report RR-90-07, Department of Data Theory, University of Leiden, Leiden, The Netherlands, 1990.

P. Verboon. *A Robust Approach to Nonlinear Multivariate Analysis*. PhD thesis, University of Leiden, 1994. Also published by DSWO Press.

H. Voss and U. Eckhardt. Linear convergence of generalized Weiszfeld's method. *Computing*, 25:243–251, 1980.

E. Weiszfeld. Sur le point par lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematics Journal*, 43:355–386, 1937.

W.I. Zangwill. *Nonlinear Programming. A Unified Approach*. Prentice-Hall, 1969.

DEPARTMENT OF STATISTICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES
E-mail address, Jan de Leeuw: deleeuw@stat.ucla.edu

DEPARTMENT OF STATISTICS, UNIVERSITY OF MICHIGAN, ANN ARBOR
E-mail address, George Michailides: gmichail@umich.edu

Visual Pushing and Grasping (VPG) is a method for training robotic agents to learn how to plan complementary pushing and grasping actions for manipulation (e.g. for unstructured pick-and-place applications). VPG operates directly on visual observations (RGB-D images), learns from trial and error, trains quickly, and generalizes to new objects and scenarios. This repository provides PyTorch code for training and testing VPG policies with deep reinforcement learning in both simulation and real-world settings on a UR5 robot arm. This is the reference implementation for the paper: Learning Synergi Analyze data to determine if a design works as intended to change the speed or direction of an object with a push or a pull. Specific Learning Outcomes: Â· I can conduct an investigation about pushes and pulls.Â Student Pictures for Push and Pull Construction Paper Graph. Push & Pull Assessment. Student NameÂ When you are finished, create a bar graph by coloring the graph to show how many push, pull, and push and pull things you found. Push & Pull Assessment. Student Name