Embedding Graphs under Centrality Constraints

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Brian Baingana

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Professor Georgios B. Giannakis, Advisor

May 2013

Acknowledgments

First, my deepest gratitude goes to Prof. Georgios B. Giannakis for giving me the opportunity to be a part of the prestigious SPiNCOM research group. His guidance and belief in me have been of such great value, making the completion of this thesis a reality. I would like to thank him for introducing me to exciting areas of research that are poised to play a big role in the technology of tomorrow. In addition, his encouragement towards clear thinking and effective communication continues to guide me along my path of illumination.

In a special way I would like to extend my appreciation to Prof. Nikos Sidiropoulos and Prof. George Karypis who agreed to serve on my thesis committee. Thanks go to a number of other professors in the departments of Electrical Engineering and Computer Science whose graduate level courses opened up my mind and prepared me to embark on this area of research.

The content of this thesis has benefited from numerous discussions with current and former members of the SPiNCOM group: Dr. Pedro Forero, Dr. Emiliano Dall' Anese, Prof. Ketan Rajawat, Dr. Vassilis Kekatos, Dr. Seung-Jun Kim, Dr. Danielle Angelosante, Dr. Eric Msechu, Dr. Alfonso Cano, Prof. Ioannis Schizas, Dr. Shahrokh Farahmand, Dr. Nikos Gatsis, Prof. Antonio Marques, Dr. Konstantinos Slavakis, Dr. Gonzalo Mateos, Prof. Hao Zhu, Dr. Juan Andres Bazerque, Morteza Mardani, Nasim Yahya Soltani, and Yu Zhang.

I am very grateful to my family and friends back home in Uganda who have been a constant source of encouragement even during the toughest of times. Foremost, my fiancee Paula Tusiime for her love, care, unwavering support, and belief in me. I highly appreciate the encouragement I have received over the years from Uncle Ezra and Auntie Jo propelling me to do my best and never limit myself in anyway. I would like to thank my siblings, Wycliffe and Patricia, for standing by my side through all the times of triumph and those of despair. Last but not least, I appreciate all the great memories enjoyed with the family of Scott and Dee McLellan, friends I have made in Minnesota, who have made my life beyond academia a wonderful experience.

Brian Baingana, Minneapolis, May 14, 2013

Abstract

Visual rendering of graphs is a key task in the mapping of complex network data. Although most graph drawing algorithms emphasize aesthetic appeal, certain applications such as travel-time maps place more importance on visualization of structural network properties. This thesis advocates two graph embedding approaches with centrality considerations to comply with node hierarchy. The embedding problem is formulated first as one of constrained multi-dimensional scaling (MDS), and it is solved via block coordinate descent iterations with successive approximations and guaranteed convergence to a *Karush-Kuhn-Tucker* (KKT) point. In addition, a regularization term enforcing graph smoothness is incorporated with the goal of reducing edge crossings. A second approach leverages the locally-linear embedding (LLE) algorithm which assumes that the graph encodes data sampled from a low-dimensional manifold. Closed-form solutions to the resulting centrality-constrained optimization problems are determined yielding meaningful embeddings. Experimental results demonstrate the efficacy of both approaches, especially for visualizing large networks on the order of thousands of nodes.

Contents

List of Figures

Chapter 1

Introduction

Graphs are a versatile tool for encoding different kinds of network data arising in a broad class of complex systems. The increasing availability of cheap computing power and storage for large-scale data has led to a "network-centric" view of systems ranging from online social relationships between people and communications infrastructures that integrate millions of computers to interactions among thousands of genes and metabolites present in living cells [1], [2]. Indeed, the analysis of networks underpins some of the most disruptive innovations over the last 15 years e.g., Google, Facebook, and Twitter.

Given a graph whose vertices represent entities in a system (e.g., web pages or friends in a social network), and edges capture interactions between them (e.g., links between web pages or friendships in social networks), graph embedding associates each vertex with a set of coordinates in low-dimensional Euclidean space. In addition to network visualization, graph embedding algorithms compress network data and provide a way to geometrically solve some computationally hard problems.

Despite the abundance of statistical analysis techniques, network visualization remains a powerful tool for capturing patterns that may not be apparent in large-scale systems. Indeed, the rising complexity and volume of networked data presents new opportunities and challenges for visual analysis tools that capture global patterns and use visual metaphors to convey meaningful structural information like hierarchy, similarity and natural communities [13].

1.1 Context and motivation

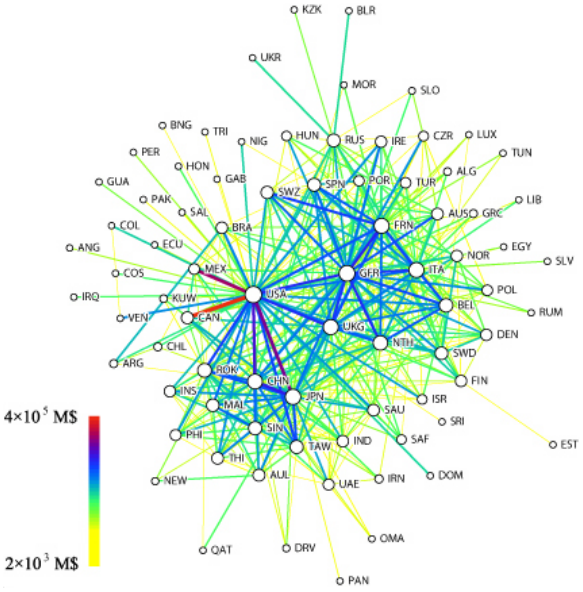
Most traditional visualization algorithms trade-off the clarity of structural characteristics of the underlying data for aesthetic requirements like minimal edge crossing and fixed inter-node distance; e.g., [6, 8, 14–20]. Although efficient for relatively small graphs (hundreds of nodes), embeddings for larger graphs using these techniques are seldom structurally informative. To this end, several approaches have been developed for embedding graphs while preserving specific structural properties.

Pioneering methods (e.g. [33]) on this front have resorted to MDS, which seeks a low-dimensional depiction of high-dimensional data in which pairwise distances between embedding coordinates are close (in a well-defined sense) to the dissimilarities between the original data points [31, 38]. In this case, the vertex dissimilarity structure is preserved through pairwise distance metrics between vertices.

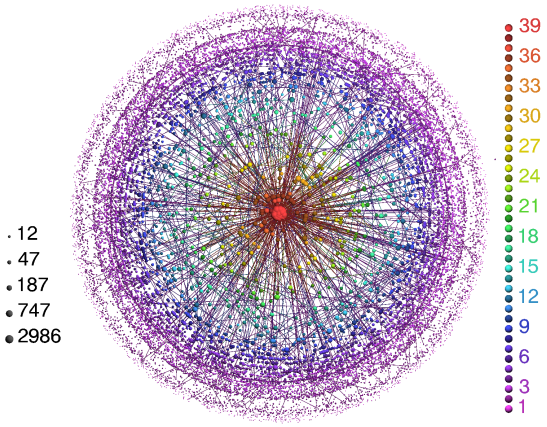
Principal component analysis (PCA) of the graph adjacency matrix is advocated in [3], leading to a spectral embedding whose vertices correspond to entries of the leading component vectors. The structure preserving embedding algorithm in [4] solves a semidefinite program with linear topology constraints so that a nearest neighbor algorithm can recover the graph edges from the embedding.

Visual analytics based approaches developed in [7] and [12] emphasize community structures with applications to community browsing in graphs. Concentric graph layouts developed in [39] and [30] capture notions of node hierarchy by placing the highest ranked nodes at the center of the embedding.

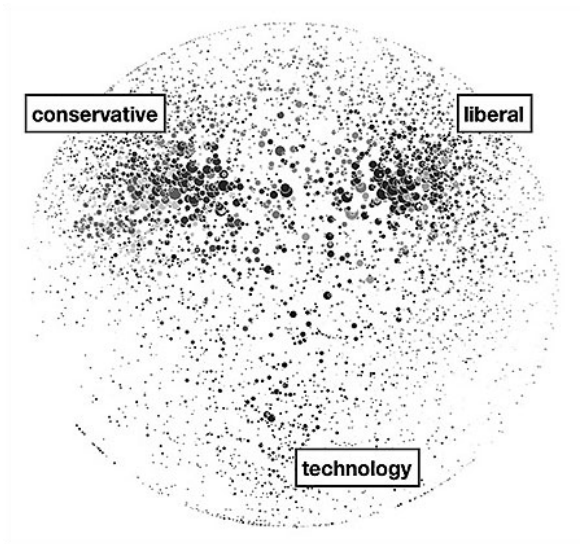
Visualization examples drawn from known networks are depicted in Figure 1.1. Structural properties in these examples are encoded by variable node sizes, edge thickness and color, as well as distance from the center of the drawing. The visualization of the global finance network in Figure 1.1a) places nodes with high degree closer to the center while Figure 1.1b) emphasizes the *coreness* of Internet nodes at the resolution of autonomous systems (AS). On the other hand, depiction of the English language "blogosphere" in Figure 1.1c) captures natural clusters present in the blog network, while Figure 1.1d) encodes the level of university influence by the node size.



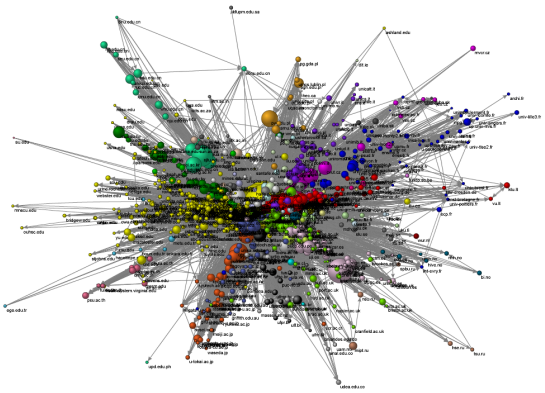
a) Global finance network



b)AS-level Internet map



c) English language blogosphere



d)World-class universities on the web

Figure 1.1: Visualization of different networks emphasizing varying properties of the networks.

From these examples, it is clear that hierarchy plays an increasingly important analytical role as the size of the networks of interest grows. Although the graph embedding problem

has been studied for years, development of fast and optimal visualization algorithms with hierarchical constraints is challenging and existing methods typically resort to heuristic approaches. The growing interest in analysis of very large networks has prioritized the need for effectively capturing hierarchy over aesthetic appeal in visualization. For instance, a hierarchy-aware visual analysis of a global computer network is naturally more useful to security experts trying to protect the most critical nodes from a viral infection. Layouts of metro-transit networks that clearly show terminals routing the bulk of traffic convey a better picture about the most critical nodes in the event of a terrorist attack.

In this thesis, hierarchy is captured through well-defined measures of node importance collectively known as *centrality* in the network science community (see e.g., [22], [23]). For instance, betweenness centrality describes the extent to which information is routed through a specific node by measuring the fraction of all shortest paths traversing it; see e.g., [1, p. 89]. Other measures include closeness, eigenvalue, and Markov centrality. Motivated by the desire for more effective approaches to capture centrality structure, dimensionality reduction techniques are leveraged to develop graph embedding algorithms that adhere to hierarchy.

1.2 Thesis contributions

This thesis builds upon dimensionality reduction approaches from machine learning to develop algorithms for embedding graphs with centrality structure captured through appropriately chosen constraints. First, MDS is considered with centrality constraints that place higher centrality nodes closer to the origin of the embedding. The resulting non-convex optimization problem is solved by recognizing a special separable structure and introducing convenient relaxations that lead to a fast algorithm converging linearly with the size of the network.

Nonlinear manifold learning algorithms [34–36, 43] preserve local distances described by a connectivity graph extracted from input data. Among these, LLE is attractive for large datasets because of the low computational cost requiring simple closed form expressions and a single spectral decomposition [34]. Although popular in machine learning [34] [36] [37], this advantage has not been exploited for large-scale network visualization applications.

To this end, LLE is leveraged for graph embedding by introducing centrality constraints. Neighborhoods are selected naturally by taking nodes within a small number of hops from each node. The centrality-constrained LLE problem yields simple closed-form solutions of low dimensionality making it competitive for embedding very large graphs.

1.3 Related work

Although the graph embedding problem has generally been studied for sometime, the use of centralities to guide visualization is recent and has been addressed predominantly in the context of scale-free networks. The derivatives of centrality metrics [5] are used to visualize social networks in a way that captures the sensitivity of nodes to changes in degree of a given node. Edge filtering based on betweenness centrality is employed by [9] to improve the layout of large graphs. Similar filtering approaches in [10] and [11] remove edges with high betweenness centrality leading to clearer layouts capturing the network structure.

Closer to the objectives of this thesis is work involving radial visualizations [30] generated by iteratively solving an MDS stress function with centrality-based weights. However, the proposed algorithm is sensitive to initialization and offers no convergence guarantee, a limitation that is addressed in this thesis. In another study whose results have been used extensively for Internet visualization, the k-core decomposition is used to hierarchically place nodes within concentric "onion-like" shells [39]. Although effective for large-scale networks, it is a heuristic method with no optimality associated with it.

1.4 Thesis outline

The rest of this thesis is organized as follows. Chapter 2 gives a brief background of the notions of network centrality and node dissimilarities pertinent to the proposed algorithms. Chapter 3 develops MDS-based approaches for the centrality-constrained graph embedding problem and Chapter 4 solves the same problem via LLE. Finally, concluding remarks and a discussion of future directions are given in Chapter 5.

1.5 Notational conventions

Upper (lower) bold face letters denote matrices (column vectors) and calligraphic letters denote sets; $\mathbf{X}^T$ denotes transpose of matrix $\mathbf{X}$, and $\mathbf{X}^\dagger$ denotes its *Penrose-Moore pseudoinverse*; $\mathrm{Tr}(\mathbf{X})$ is the trace operator of matrix $\mathbf{X}$; $\|\mathbf{x}\|_2$ is the Euclidean norm of $\mathbf{x}$; $\mathbf{e}_i$ is the i-th column of the identity matrix, $\mathbf{I}$. $\mathbf{X}_{\mathcal{I},\mathcal{J}}$ is the submatrix of $\mathbf{X}$ obtained by retaining the common elements in rows indexed by the elements of $\mathcal{I}$ and columns indexed by $\mathcal{J}$. Lastly, $\partial f(.)$ denotes the subdifferential of $f(.)$.

Chapter 2

Background

From an unsupervised machine learning perspective, graph embedding amounts to a dimensionality reduction problem which has been extensively studied and benefits from a plethora of well-documented methods. This chapter briefly establishes this link and introduces the notions of network centrality and node dissimilarity that will play a key role in the development of the proposed graph embedding approaches.

2.1 Embedding graphs by dimensionality reduction

Given a set of N data points in q-dimensional Euclidean space, i.e., $\{\mathbf{y}_n \in \mathbb{R}^q\}_{n=1}^N$, dimensionality reduction seeks the points $\{\mathbf{x}_n \in \mathbb{R}^p\}_{n=1}^N$ where $p < q$. In the graph embedding context, each of the N points is associated with a node and the topology is captured by the $N \times N$ adjacency matrix, $\mathbf{A}$, whose non-zero entries denote edge weights. Dimensionality reduction in this case amounts to determining an $N \times p$ matrix whose rows correspond to lower-dimensional renditions of the nodes.

Multi-dimensional scaling (MDS) recovers low dimensional vectors, $\{\mathbf{x}_n\}_{n=1}^N$, so that the resulting pairwise Euclidean distance, $\|\mathbf{x}_n - \mathbf{x}_m\|_2$, is representative of the dissimilarity δ_{nm}, between $\mathbf{y}_n$ and $\mathbf{y}_m$ [31]. In the classical setup it is assumed that $\delta_{nm} = \|\mathbf{y}_n - \mathbf{y}_m\|_2$, and MDS does not require explicit knowledge of $\{\mathbf{y}_n\}_{n=1}^N$. The sole dependence on pairwise dissimilarities places MDS at the forefront of a number of successful graph

embedding algorithms e.g., [30, 33]. This is because graph-theoretic distances (e.g., the shortest path distance) are more representative of the underlying topology than Euclidean distances between the rows of $\mathbf{A}$.

Spectral embedding determines $\{\mathbf{x}_n\}_{n=1}^N$ by applying PCA to $\mathbf{A}$, retaining only the p leading eigenvectors. More recently, Laplacian eigenmaps offer a compelling alternative to PCA by spectral decomposition of the graph Laplacian $\mathbf{L} := \mathbf{D} - \mathbf{A}$ where $\mathbf{D} := \text{diag}(\mathbf{A1})$, and retaining the p leading eigenvectors [32]. In general, PCA and MDS are limited by their linear nature i.e., they assume that the high dimensional data are sampled from a linear subspace. However, in many settings the data lie close to a low-dimensional nonlinear manifold and linear methods fail to utilize this knowledge for improved prediction accuracy.

To this end, manifold learning approaches such as LLE [34] and Hessian LLE [38] have gained popularity in application domains like computer vision and image processing. For instance, LLE assumes that the underlying manifold structure is locally linear and fits each datum to a linear combination of its neighbors in order to determine the unknown reconstruction weights by least-squares. The sought embedding vectors are recovered by a similar fitting procedure that preserves the reconstruction weights per neighborhood. The neigborhood-preserving properties of non-linear embedding approaches implicitly emphasize natural structural properties like clustering of the data.

In order to adapt MDS and LLE to the centrality-constrained graph embedding task, it is assumed that the network centralities and node dissimilarities are known inputs. However, assuming complete knowledge of the graph, both inputs can be efficiently computed as part of a pre-processing step. The proposed approaches then apply well-defined centrality constraints to the dimensionality reduction criteria yielding the sought embeddings. The remainder of this chapter defines a few of the common centrality and dissimilarity metrics that will be referred to throughout the rest of this thesis.

2.2 Network centrality

Centrality measures offer a valuable means of quantifying the level of importance attributed to a specific node or edge within the network. For instance, such measures address questions

pertaining to the most authoritative authors in a research community, the most influential web pages, or which genes would be most lethal to an organism if deleted. Such questions often arise in the context of social, information and biological networks and a number of centrality measures exist ranging from the straightforward ones like node degree to more sophisticated ones like PageRank [28], the web ranking methodology originally developed for the Google search engine. A brief description of the common metrics follows.

2.2.1 Degree centrality

Degree centrality refers to the number of edges adjacent to a particular node. In directed graphs, specifying the in-degree and out-degree corresponding to the direction of the edged is more meaningful. A high degree is usually associated with more influence within the neighborhood of the node.

2.2.2 Closeness centrality

Closeness centrality captures the extent to which a particular node is close to many other nodes in the network. Although several methods are available, the standard approach determines the inverse of the total geodesic distance between the node whose closeness centrality is desired and all the other nodes in the network [22]. If c_i^{cl} denotes the closeness centrality of node i, then

$$c_i^{cl} := \frac{1}{\sum_{j \in \mathcal{V}} d_{ij}} \tag{2.1}$$

where d_{ij} is the geodesic distance (lowest sum of edge weights) between nodes i and j. In the context of visualization requirements for large social networks, an embedding which encodes the level of influence of a paper in a citation network or an autonomous system within the internet would place nodes with highest closeness centrality at the center of the embedding.

2.2.3 Betweenness centrality

Betweenness centrality, on the other hand, summarizes the extent to which nodes are located between other pairs of nodes. It measures the fraction of shortest paths that traverse a node

to all shortest paths in the network and is commonly defined as

$$c_i^b := \frac{\sum_{j \neq k \neq i \in \mathcal{V}} \sigma_{j,k}^i}{\sum_{i \in \mathcal{V}} \sigma_{j,k}^i} \tag{2.2}$$

where $\sigma_{j,k}^i$ is the number of shortest paths between nodes j and k through node i [23]. An immediate application of betweenness centrality to network visualization is envisaged in the context of exploratory analysis of transport networks (for instance a network whose nodes are transit and terminal stations and edges represent connections by rail lines). In such a network, a knowledge of stations that route the most traffic is critical for transit planning and visualizations that effectively capture the betweenness centrality are well motivated.

2.3 Node dissimilarities

This section briefly discusses some of the dissimilarity measurements crucial for the graph embedding task. Although the set of nodes can represent several inter-related objects with well defined criteria for pairwise dissimilarities (e.g., $\delta_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|_2$, where vectors $\mathbf{y}_i$ and $\mathbf{y}_j$ corresponding to nodes i and j are available in Euclidean space, and δ_{ij} denotes the dissimilarity between the two nodes), dissimilarities must be determined entirely from the graph in order to obtain embeddings that reliably encode the underlying network topology. Node dissimilarity metrics can be broadly classified as: i) methods based on the number of shared neighbors, and ii) methods based on graph-theoretic distances.

2.3.1 Shared neighbors

Graph partitioning techniques tend to favor dissimilarities based on the number of shared neighbors [1]. Consider for instance

$$\delta_{ij} = \frac{|\mathcal{N}_i \triangle \mathcal{N}_j|}{d_{(|V|)} + d_{(|V|-1)}} \tag{2.3}$$

where $\mathcal{A} \triangle \mathcal{B}$ denotes the symmetric difference between sets $\mathcal{A}$ and $\mathcal{B}$; and $d_{(i)}$ denotes the i-th smallest element in the degree sequence of $\mathcal{G}$. The ratio in (2.3) yields a normalized metric over $[0, 1]$, and computes the number of single-hop neighbors of i and j that are

not shared. An alternative measure commonly used in hierarchical clustering computes the Euclidean distance between rows of the adjacency matrix, namely

$$\delta_{ij} = \left(\sum_{k=1}^{N} (a_{ik} - a_{jk})^2 \right)^{\frac{1}{2}} \tag{2.4}$$

where $\mathbf{a}_i$ denotes the i-th row of the adjacency matrix $\mathbf{A}$, and $\mathbf{a}_i := [a_{i1}, \ldots, a_{iN}]^T$. For unweighted graphs and ignoring normalization, (2.3) is equivalent to the square of (2.4) since $|\mathcal{N}_i \triangle \mathcal{N}_j| = \|\mathbf{a}_i \oplus \mathbf{a}_j\|_1$. A downside to these metrics is their tendency to disproportionately assign large dissimilarities to pairs of highly connected nodes that do not share single-hop neighbors.

2.3.2 Graph-theoretic distances

Graph-theoretic distances offer a more compelling option for visualization requirements because of their global nature and ability to capture meaningful relationships between nodes that do not share immediate neighbors. The *shortest-path* distance is popular and has been used for many graph embedding approaches because of its simplicity and the availability of algorithms for its determination. It is important to note that it assigns the same distance to node pairs separated by multiple shortest paths as those with a single shortest path yet intuition suggests that node pairs separated by multiple shortest paths are more strongly connected and therefore more similar.

This shortcoming is alleviated by dissimilarity (and similarity) measures based on Markov-chain models of random-walks, which have become increasingly important for tasks such as collaborative recommendation (see e.g., [40]). Nodes are viewed as states and a transition probability, $p_{ij} = a_{ij}/(\sum_{j=1}^{N} a_{ij})$, is assigned to each edge. The *average commute time* is defined as the quantity $n(i,j) = m(j|i) + m(i|j)$, where $m(j|i)$ denotes the *average first-passage time*, and equals the average number of hops taken by a random walker to reach node j given that the starting node was i. Although $n(i,j)$ is a recursion defined in terms of transition probabilities, it admits the following closed form [40]

$$n(i,j) = \Omega \left(\mathbf{e}_i - \mathbf{e}_j \right)^T \mathbf{L}^\dagger \left(\mathbf{e}_i - \mathbf{e}_j \right) \tag{2.5}$$

where $\Omega = \sum_{i,j} a_{ij}$, and $\mathbf{L}$ denotes the graph Laplacian matrix. Moreover, taking the spectral decomposition, $\mathbf{L}^{\dagger} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{T}$, and expressing the canonical basis vector, $\mathbf{e}_i$, as $\mathbf{e}_i = \mathbf{U}\mathbf{z}_i$ with $\mathbf{z}'_i := \mathbf{\Lambda}^{1/2}\mathbf{z}_i$, yields

$$[n(i,j)]^{1/2} = \Omega\|\mathbf{z}'_i - \mathbf{z}'_j\|_2. \tag{2.6}$$

Evidently, $[n(i,j)]^{1/2}$ is an Euclidean distance in the space spanned by the vectors $\{\mathbf{e}_i\}_{i=1}^{N}$, each corresponding to a node in $\mathcal{G}$, and is known as the *Euclidean commute-time distance (ECTD)* (see [40]). For the graph embedding task, setting $\delta_{ij} = [n(i,j)]^{1/2}$ is convenient because $\mathcal{G}$ is treated as N-dimensional Euclidean space with a well-defined criterion for measuring pairwise dissimilarities.

Chapter 3

Centrality-constrained MDS

MDS seeks a low-dimensional depiction of high-dimensional data in which pairwise Euclidean distances between embedding coordinates are close (in a well-defined sense) to the dissimilarities between the original data points [31], [38]. Closeness criteria (a.k.a. stress costs [31, Chap. 3]) are generally non-convex, and the quest for global optimality is challenging because ordinary descent methods do not have optimality guarantees, and are sensitive to initialization. Successive approximation with global and convex upper bounds is used in [31, Chap. 8] to minimize the stress cost yielding near-optimal results.

This chapter incorporates centrality by adopting an MDS criterion with radial constraints that place nodes of higher centrality closer to the origin of the graph embedding.The novel approach exploits the block separability inherent to the proposed model and adapts the coordinate descent algorithm to determine the optimal embedding. Edge crossings are minimized by regularizing the cost with a smoothness promoting term weighted by a tuning parameter. Smoothness encourages nodes that share an edge to lie closer to each other in the embedding. As a result, the length and hence the number of edge crossings in the network visualization is markedly reduced. In addition, the regularization term offers the benefit of incorporating the underlying network topology when the dissimilarities considered are not graph-theoretic e.g., Euclidean distances between feature vectors assigned to each node. Moreover, numerical tests illustrate that judicious selection of the tuning parameter results in fewer block coordinate descent (BCD) iterations, which in turn yields a visually

appealing embedding.

3.1 Problem statement

Consider a network represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ denotes the set of edges, and $\mathcal{V}$ the set of vertices with cardinality $|\mathcal{V}| = N$. It is assumed that $\mathcal{G}$ is undirected, unweighted and has no loops or multi-edges between node pairs. Let δ_{ij} denote the real-valued pairwise dissimilarity between two nodes i and j computed via a well-defined criterion. The dissimilarities are assumed to satisfy the following properties: (i) $\delta_{ij} \geq 0$; (ii) $\delta_{ij} = \delta_{ji}$; and, (iii) $\delta_{ii} = 0$. Consequently, it suffices to know the set $\mathcal{D} := \{\{\delta_{ij}\}_{j=1}^{N}\}_{i=j+1}^{N}$ with cardinality $|\mathcal{D}| = N(N-1)/2$. Let c_i denote a centrality measure assigned to node i, and consider the set $\mathcal{C} := \{c_i\}_{i=1}^{N}$ with cardinality $|\mathcal{C}| = N$.

Given $\mathcal{G}$, $\mathcal{D}$, $\mathcal{C}$, and the prescribed embedding dimension p (typically $p \in \{2, 3\}$), the graph embedding task amounts to finding the set of $p \times 1$ vectors $\mathcal{X} := \{\mathbf{x}_i\}_{i=1}^{N}$ which "respect" the network structure characteristics encoded through $\mathcal{D}$ and $\mathcal{C}$. MDS amounts to finding vectors $\{\mathbf{x}_i\}_{i=1}^{N}$ so that the embedding coordinates $\mathbf{x}_i$ and $\mathbf{x}_j$ satisfy $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \approx \delta_{ij}$ by solving the following problem:

$$\text{(P0)} \quad \hat{\mathcal{X}} = \underset{\mathbf{x}_1,\ldots,\mathbf{x}_N}{\arg\min} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} [\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \delta_{ij}]^2. \tag{3.1}$$

Centrality structure will be imposed on (3.1) by constraining $\mathbf{x}_i$ to have a centrality-dependent radial distance $f(c_i)$, where $f(.)$ is a monotone decreasing function. For instance, $f(c_i) = \alpha e^{-\beta c_i}$ where α and β are positive constants. The resulting constrained optimization problem now becomes

$$\begin{aligned} \text{(P1)} \quad \hat{\mathcal{X}} = \underset{\mathbf{x}_1,\ldots,\mathbf{x}_N}{\arg\min} \quad & \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \delta_{ij}\right]^2 \\ \text{s. to} \quad & \|\mathbf{x}_i\|_2 = f(c_i),\ i = 1, \ldots, N. \end{aligned} \tag{3.2}$$

Although P0 is non-convex, standard solvers rely on gradient descent iterations but have no guarantees of convergence to the global optima [49]. Lack of convexity is exacerbated in P1 by the non-convex constraint set rendering its solution even more challenging than

that of P0. However, considering a single embedding vector $\mathbf{x}_i$, and fixing the rest $\{\mathbf{x}_j\}_{j\neq i}$, the constraint set simplifies to $\|\mathbf{x}_i\|_2 = f(c_i)$, for which an appropriate relaxation can be sought. Key to the algorithm proposed next lies in this inherent decoupling of the centrality constraints.

Remark 3.1. *A number of stress costs with well-documented merits have been reported for MDS. The choice made in P*0 *is motivated by convenience of adaptation to BCD iterations discussed in Section 3.2. Moreover, more general forms include weights, which have been ignored without loss of generality.*

3.2 BCD with successive approximations

By exploiting the separable nature of the cost as well as the norm constraints in (3.2), BCD will be developed in this section to arrive at a solution approaching the global optimum. To this end, the centering constraint $\sum_{i=1}^{N}\mathbf{x}_i = \mathbf{0}$, typically invoked to fix the inherent translation ambiguity, will be dropped first so that the problem remains decoupled across nodes. The effect of this relaxation can be compensated for by computing the centroid of the solution of (3.2), and subtracting it from each coordinate. The N equality norm constraints are also relaxed to $\|\mathbf{x}_i\|_2 \leq f(c_i)$. Although the entire constraint set is non-convex, each relaxed constraint is a convex and closed Euclidean ball with respect to each node in the network.

Let $\mathbf{x}_i^r$ denote the minimizer of the optimization problem over block i, when the remaining blocks $\{\mathbf{x}_j\}_{j\neq i}$ are fixed during the BCD iteration r. By fixing the blocks $\{\mathbf{x}_j\}_{j\neq i}$ to their most recently updated values, the sought embedding is obtained as

$$\begin{aligned} \hat{\mathbf{x}}_i &= \arg\min_{\mathbf{x}} \frac{1}{2}\sum_{j\neq i}\left[\|\mathbf{x}-\mathbf{x}_j\|_2 - \delta_{ij}\right]^2 \\ \text{s. to} \quad & \|\mathbf{x}\|_2 \leq f(c_i) \end{aligned} \tag{3.3}$$

or equivalently as

$$\begin{aligned} \arg\min_{\mathbf{x}} \quad & \frac{(N-1)}{2}\|\mathbf{x}\|_2^2 - \mathbf{x}^T\Big(\sum_{j<i}\mathbf{x}_j^r + \sum_{j>i}\mathbf{x}_j^{r-1}\Big) \\ & - \sum_{j<i}\delta_{ij}\|\mathbf{x}-\mathbf{x}_j^r\|_2 - \sum_{j>i}\delta_{ij}\|\mathbf{x}-\mathbf{x}_j^{r-1}\|_2 \\ \text{s. to} \quad & \|\mathbf{x}\|_2 \leq f(c_i) \end{aligned} \tag{3.4}$$

where $\sum_{j<i}(.) := \sum_{j=1}^{i-1}(.)$ and $\sum_{j>i}(.) := \sum_{j=i+1}^{N}(.)$. With the last two sums in the cost function of (3.4) being non-convex and non-smooth, convergence of the BCD algorithm cannot be guaranteed [45, p. 272]. Moreover, it is desired to have each per-iteration subproblem solvable to global optimality, in closed form and at a minimum computational cost. The proposed approach seeks a global upper bound of the objective with the desirable properties of smoothness and convexity. To this end, consider the function $\Psi(\mathbf{x}) := \psi_1(\mathbf{x}) - \psi_2(\mathbf{x})$, where

$$\psi_1(\mathbf{x}) := \frac{(N-1)}{2}\|\mathbf{x}\|_2^2 - \mathbf{x}^T\Big(\sum_{j<i}\mathbf{x}_j^r + \sum_{j>i}\mathbf{x}_j^{r-1}\Big) \tag{3.5}$$

and

$$\psi_2(\mathbf{x}) := \sum_{j<i}\delta_{ij}\|\mathbf{x}-\mathbf{x}_j^r\|_2 + \sum_{j>i}\delta_{ij}\|\mathbf{x}-\mathbf{x}_j^{r-1}\|_2. \tag{3.6}$$

Note that $\psi_1(\mathbf{x})$ is a convex quadratic function, and that $\psi_2(\mathbf{x})$ is convex (with respect to $\mathbf{x}$) but non-differentiable. The first-order approximation of (3.6) at any point in its domain is a global under-estimate of $\psi_2(\mathbf{x})$. Despite the non-smoothness at some points, such a lower bound can always be established using its subdifferential. As a consequence of the convexity of $\psi_2(\mathbf{x})$, it holds that [45, p. 731]

$$\psi_2(\mathbf{x}) \geq \psi_2(\mathbf{x}_0) + \mathbf{g}^T(\mathbf{x}_0)(\mathbf{x}-\mathbf{x}_0), \forall \mathbf{x} \in \mathrm{dom}(\psi_2) \tag{3.7}$$

where $\mathbf{g}(\mathbf{x}) \in \partial\psi_2(\mathbf{x})$ is a subgradient within the subdifferential set, $\partial\psi_2(\mathbf{x})$ of $\psi_2(\mathbf{x})$. The subdifferential of $\|\mathbf{x}-\mathbf{x}_j\|_2$ with respect to $\mathbf{x}$ is given by

$$\partial_{\mathbf{x}}\|\mathbf{x}-\mathbf{x}_j\|_2 = \begin{cases} \frac{\mathbf{x}-\mathbf{x}_j}{\|\mathbf{x}-\mathbf{x}_j\|_2}, & \text{if } \mathbf{x} \neq \mathbf{x}_j \\ \mathbf{s} \in \mathbb{R}^p : \ \|\mathbf{s}\|_2 \leq 1, & \text{otherwise} \end{cases} \tag{3.8}$$

which implies that

$$\partial_{\mathbf{x}}\psi_2(\mathbf{x}) = \sum_{j=1}^{N} \delta_{ij}\partial_{\mathbf{x}}\|\mathbf{x}-\mathbf{x}_j\|_2. \tag{3.9}$$

Using (3.7), it is possible to lower bound (3.6) by

$$\begin{aligned}\psi_2'(\mathbf{x},\mathbf{x}_0) &= \sum_{j<i}\delta_{ij}\left[\|\mathbf{x}_0-\mathbf{x}_j^r\|_2 + (\mathbf{g}_j^r)^T(\mathbf{x}_0)(\mathbf{x}-\mathbf{x}_0)\right] \\ &\quad + \sum_{j>i}\delta_{ij}\left[\|\mathbf{x}_0-\mathbf{x}_j^{r-1}\|_2 + (\mathbf{g}_j^{r-1})^T(\mathbf{x}_0)(\mathbf{x}-\mathbf{x}_0)\right]. \end{aligned}\tag{3.10}$$

Consider now $\Phi(\mathbf{x},\mathbf{x}_0) := \psi_1(\mathbf{x}) - \psi_2'(\mathbf{x},\mathbf{x}_0)$, and note that $\Phi(\mathbf{x},\mathbf{x}_0)$ is convex and globally upper bounds the cost in (3.4). The proposed BCD algorithm involves successive approximations using (3.10), and yields the following QCQP per block

$$\text{(P2)} \qquad \underset{\{\mathbf{x}:\|\mathbf{x}\|_2\le f(c_i)\}}{\arg\min}\ \Phi(\mathbf{x},\mathbf{x}_0). \tag{3.11}$$

For convergence, $\mathbf{x}_0$ must be selected to satisfy the following conditions [46]:

$$\Phi(\mathbf{x}_0,\mathbf{x}_0) = \Psi(\mathbf{x}_0), \quad \forall \mathbf{x}_0 \in \mathcal{F} \tag{3.12a}$$

$$\Phi(\mathbf{x},\mathbf{x}_0) \ge \Psi(\mathbf{x}), \quad \|\mathbf{x}\|_2 \le f(c_i), \forall i \tag{3.12b}$$

where $\mathcal{F} := \bigcup_{i=1}^{N}\{\mathbf{x} : \|\mathbf{x}\|_2 \le f(c_i)\}$. In addition, $\Phi(\mathbf{x},\mathbf{x}_0)$ must be continuous in $(\mathbf{x},\mathbf{x}_0)$.

Proposition 3.1. *The conditions for convergence in (3.12a) and (3.12b) are satisfied by selecting* $\mathbf{x}_0 = \mathbf{x}^{r-1}$.

The proof of Proposition 1 involves substituting $\mathbf{x}_0 = \mathbf{x}^{r-1}$ in (3.12a) and (3.12b) (see Appendix A). Taking successive approximations around $\mathbf{x}^{r-1}$ in P2, ensures the uniqueness of

$$\mathbf{x}_i^r = \underset{\{\mathbf{x}:\|\mathbf{x}\|_2\le f(c_i)\}}{\arg\min}\ \frac{(N-1)}{2}\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\left[\sum_{j<i}(\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}^{r-1})) + \sum_{j>i}(\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}^{r-1}))\right]. \tag{3.13}$$

Solving (3.13) amounts to obtaining the solution of the unconstrained QP, $(\mathbf{x}^*)^r$, and projecting it onto $\{\mathbf{x} : \|\mathbf{x}\|_2 \le f(c_i)\}$; that is,

$$\mathbf{x}_i^r = \begin{cases} \frac{(\mathbf{x}^*)^r}{\|(\mathbf{x}^*)^r\|_2} f(c_i), & \text{if} \quad \|(\mathbf{x}^*)^r\|_2 > f(c_i) \\ (\mathbf{x}^*)^r, & \text{otherwise} \end{cases} \tag{3.14}$$

where

$$(\mathbf{x}^*)^r = \frac{1}{N-1}\left[\sum_{j<i}(\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}^{r-1})) + \sum_{j>i}(\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}^{r-1}))\right]. \tag{3.15}$$

It is desirable but not necessary that the algorithm converges because depending on the application, reasonable network visualizations can be found with fewer iterations. In fact, successive approximations merely provide a more refined graph embedding that maybe more aesthetically appealing.

Although the proposed algorithm is guaranteed to converge, the solution is only unique up to a rotation and a translation (cf. MDS). In order to eliminate the translational ambiguity, the embedding can be centered at the origin. Assuming that the optimal blocks determined within outer iteration r are reassembled into the embedding matrix $\mathbf{X}^r := \left[(\mathbf{x}_1^r)^T, \ldots, (\mathbf{x}_N^r)^T\right]^T$, the final step involves subtracting the mean from each coordinate using the centering operator as follows, $\mathbf{X} = (\mathbf{I} - N^{-1}\mathbf{1}\mathbf{1}^T)\mathbf{X}^r$, where $\mathbf{I}$ denotes the $N \times N$ identity matrix, and $\mathbf{1}$ is the $N \times 1$ vector of all ones.

The novel graph embedding scheme is summarized as Algorithm 1 with matrix $\mathbf{\Delta}$ having (i,j)th entry the dissimilarity δ_{ij}, and ϵ denoting a tolerance level.

3.3 Enforcing graph smoothness

In this section, the MDS stress in (3.3) is regularized through an additional constraint that encourages smoothness over the graph. Intuitively, despite the requirement that the node placement in low-dimensional Euclidean space respects inherent network structure, through preserving e.g., node centralities, neighboring nodes in a graph-theoretic sense (meaning nodes that share an edge) are expected to be close in Euclidean distance within the embedding. An example of an application where smoothness is well motivated is the visualization of transportation networks that are defined over geographical regions. Despite the interest in their centrality structure, embeddings that place nodes that are both geographically close and adjacent in the representative graph are more acceptable to users that are familiar with their conventional (and often geographically representative) maps. Such a requirement can be captured by incorporating a constraint that discourages large distances between neigh-

Algorithm 1 BCD with successive approximations

Input: $\{c_i\}_{i=1}^N$, $\mathbf{\Delta}$, ϵ
Initialize $\mathbf{X}^0$, $r = 0$
repeat
  $r = r + 1$
  for $i = 1 \dots N$ **do**
    $(\mathbf{x}^*)^r = \frac{1}{N-1}[\sum_{j<i}(\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}^{r-1})) + \sum_{j>i}(\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}^{r-1}))]$
    if $\|(\mathbf{x}^*)^r\|_2 > f(c_i)$ **then**
      $\mathbf{x}_i^r = \frac{(\mathbf{x}^*)^r}{\|(\mathbf{x}^*)^r\|_2} f(c_i)$
    else
      $\mathbf{x}_i^r = (\mathbf{x}^*)^r$
    end if
    $\mathbf{X}^r(i,:) = (\mathbf{x}_i^r)^T$
  end for
until $\|\mathbf{X}^r - \mathbf{X}^{r-1}\|_F \leq \epsilon$
$\mathbf{X} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbf{X}^r$

boring nodes. In essence, this constraint enforces smoothness over the graph embedding.

A popular choice of a smoothness-promoting function is $h(\mathbf{X}) := \mathrm{Tr}(\mathbf{X}^T\mathbf{L}\mathbf{X})$, where $\mathrm{Tr}(.)$ denotes the trace operator. It can be shown that $h(\mathbf{X}) = (1/2)\sum_{i=1}^{N}\sum_{i=1}^{N} a_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$, where a_{ij} is the (i,j)th entry of $\mathbf{A}$. Motivated by penalty methods in optimization, the cost in (3.2) will be augmented as follows

$$\begin{aligned} \text{(P3)} \qquad \underset{\mathbf{x}_1,\ldots,\mathbf{x}_N}{\arg\min} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \delta_{ij}\right]^2 + \frac{\lambda}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \\ \text{s. to} \quad & \|\mathbf{x}_i\|_2 = f(c_i), i = 1,\ldots,N \end{aligned} \tag{3.16}$$

where the scalar $\lambda \geq 0$ controls the degree of smoothness. The penalty term has a separable structure and is convex with respect to $\mathbf{x}_i$. Consequently, P3 lies within the framework of successive approximations required to solve each per-iteration subproblem. Following the same relaxations and invoking the successive upper bound approximations described earlier, yields the following QCQP

$$\begin{aligned} \mathbf{x}_i^r = \underset{\{\mathbf{x}:\|\mathbf{x}\|_2 \leq f(c_i)\}}{\arg\min} \frac{(N + \lambda d_{ii} - 1)}{2}\mathbf{x}^T\mathbf{x} \quad & -\mathbf{x}^T\Bigg[\sum_{j<i}((1+\lambda a_{ij})\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}^{r-1})) \\ & + \sum_{j>i}((1+\lambda a_{ij})\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}^{r-1}))\Bigg] \end{aligned} \tag{3.17}$$

with $d_{ii} := \sum_{j=1}^{N} a_{ij}$ denoting the degree of node i.

The solution of (3.17) can be expressed as [cf. (3.14)]

$$\begin{aligned} (\mathbf{x}^*)^r = \frac{1}{N + \lambda d_{ii} - 1} \quad & \Bigg[\sum_{j<i}((1+\lambda a_{ij})\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}^{r-1})) \\ & + \sum_{j>i}((1+\lambda a_{ij})\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}^{r-1}))\Bigg]. \end{aligned} \tag{3.18}$$

With λ given, Algorithm 2 summarizes the steps to determine the constrained embedding with a smoothness penalty.

Algorithm 2 Incorporating smoothness

Input: $\mathbf{A}$, $\{c_i\}_{i=1}^N$, $\mathbf{\Delta}$, ϵ, λ
Initialize $\mathbf{X}^0$, $r = 0$
repeat
  $r = r + 1$
  for $i = 1 \dots N$ **do**
    $(\mathbf{x}^*)^r = \frac{1}{N+\lambda d_{ii}-1}[\sum_{j<i}((1+\lambda a_{ij})\mathbf{x}_j^r + \delta_{ij}\mathbf{g}_j^r(\mathbf{x}^{r-1})) + \sum_{j>i}((1+\lambda a_{ij})\mathbf{x}_j^{r-1} + \delta_{ij}\mathbf{g}_j^{r-1}(\mathbf{x}^{r-1}))]$
    if $\|(\mathbf{x}^*)^r\|_2 > f(c_i)$ **then**
      $\mathbf{x}_i^r = \frac{(\mathbf{x}^*)^r}{\|(\mathbf{x}^*)^r\|_2} f(c_i)$
    else
      $\mathbf{x}_i^r = (\mathbf{x}^*)^r$
    end if
    $\mathbf{X}^r(i,:) = (\mathbf{x}_i^r)^T$
  end for
until $\|\mathbf{X}^r - \mathbf{X}^{r-1}\|_F \leq \epsilon$
$\mathbf{X} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbf{X}^r$

3.4 Numerical experiments

In this section, the proposed algorithms are used to visualize two real-world networks that serve entirely different purposes. Appropriate centrality measures are selected to highlight particular aspects of the two networks.

3.4.1 The London tube

The first network under consideration is encoded by an undirected graph of 307 nodes representing the London tube, an underground train transit network[1]. The nodes represent stations whereas the edges represent the routes connecting them. The objective is to generate an embedding in which stations traversed by most routes are placed closer to the center, thus highlighting their relative significance in metro transit.

Such information is encoded by the betweenness centralities of the tube stations. It is worth mentioning that pursuit of this goal deviates from traditional metro transit map requirements which, although drawn using geographically inaccurate schemes, visually emphasize accessibility between adjacent stations. Figure 3.1 shows a 50-bin histogram of the betweenness centralities computed for the London tube. As expected from measurements performed on real networks, the distribution follows a power law with a small number of stations disproportionately accounting for the higher centrality values. Knowledge of the centrality distribution guides the selection of a transformation function, $f(c_i)$, that will result in a well-spaced graph embedding. Bearing this in mind, the centrality values were transformed as follows:

$$f\left(c_i\right) = \frac{\operatorname{diam}\left(\mathcal{G}\right)}{2}\left(1 - \frac{c_i - \min\limits_{i\in\mathcal{V}} c_i}{\max\limits_{i\in\mathcal{V}} c_i - \max\limits_{i\in\mathcal{V}} c_i}\right) \tag{3.19}$$

with $\operatorname{diam}(\mathcal{G})$ denoting the diameter of $\mathcal{G}$.

Figure 3.3 depicts the optimal embedding obtained without a smoothness penalty by running Algorithm 1. In order to compute the subgradients, vector $\mathbf{s}$ was set to $(1/\sqrt{2})[1,1]^T$ in (3.8). The color grading reflects the centrality levels of the nodes from highest (violet)

[1] https://wikis.bris.ac.uk/display/ipshe/London+Tube

to lowest (red). Algorithm 1 converged after approximately 150 outer iterations as shown in Figure 3.5. Turning attention to Algorithm 2, simulations were run for several values of λ starting with $\lambda = 0$, and the resultant two-dimensional ($p = 2$) embeddings were plotted as shown in Figure 3.6. Increasing λ promotes embeddings in which edge crossings are minimized. This intuitively makes sense because by forcing single-hop neighbors to lie close to each other, the average edge length decreases, leading to fewer edge crossings. In addition, increasing λ yielded embeddings that were aesthetically more appealing under fewer iterations. For instance, setting $\lambda = 10,000$ required only 30 iterations for a visualization that is comparable to running 150 iterations with $\lambda = 0$.

In order to draw comparisons with traditional network visualization techniques, the London tube transit network was drawn using the *springer-embedder* algorithm [21], an approach that falls under the broader category of force-directed visualization models. Nodes in the network are treated as balls and edges as springs. The graph embedding task amounts to determining the positions in which the balls will stabilize after the system is perturbed by stretching some of the springs. Figure 3.4 depicts the resultant *spring* layout for the London tube with the centralities encoded by the same node colors. Despite the success of force-directed methods for the visualization of small networks, it is clear that their ability to communicate meaningful structural characteristics rapidly degrades even for moderately sized networks. In contrast, the proposed approaches convey a more informative picture about the hierarchical structure of the network.

3.4.2 ArXiv general relativity (AGR) network

In this experiment, a considerably large social network is considered from the e-print arXiv repository covering scientific collaborations between authors on papers submitted to the "General Relativity and Quantum Cosmology" category (January 1993 to April 2003) [50]. The nodes represent authors and an edge exists between nodes i and j if authors i and j co-authored a paper. Although the network contains $5,242$ nodes, the embedding considered only its largest strongly connected component comprising $4,158$ nodes. The objective was to embed the network so that authors whose research is most related to the majority of the

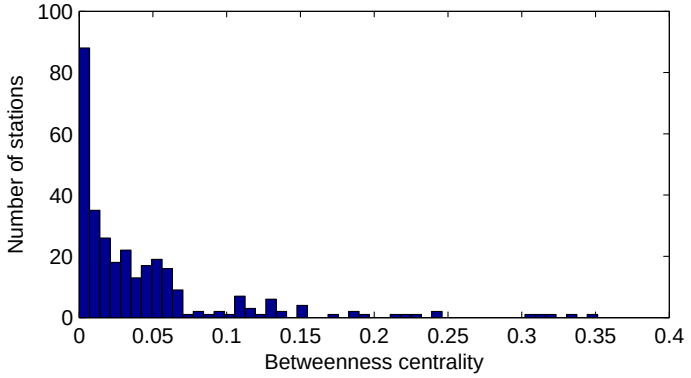


Figure 3.1: Betweenness centrality distribution of the London tube.

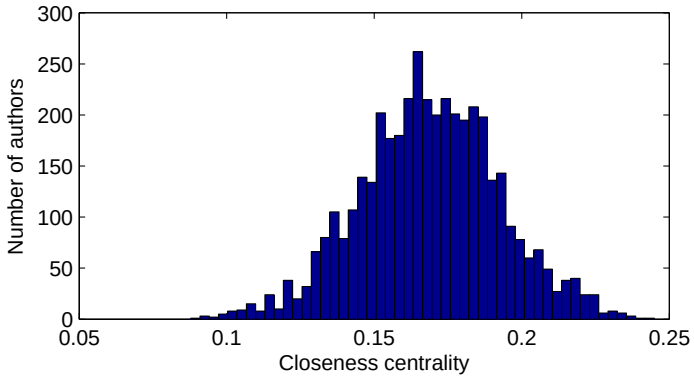


Figure 3.2: Closeness centrality distribution of the AGR network

others are placed closer to the center.

This behavior is best captured by the closeness centrality whose distribution for the AGR network is shown in Figure 3.2 (using 50 bins). In this case, the transformation $f(c_i) = \alpha e^{-\beta c_i}$, where $\alpha = 1$ and $\beta = 10$, was used. The constants α and β were chosen after trying several combinations and selecting the pair that yielded an embedding that is evenly spread out (i.e., with minimal coleascing of nodes).

Figure 3.7 shows the embedding obtained after 30 outer BCD iterations of running Algorithm 1. For clarity and emphasis of the node positions, edges were not included in the visualization. Drawings of graphs as large as the autonomous systems within the Internet typically thin out most of the edges. The color coding is adopted to reflect centrality variations.

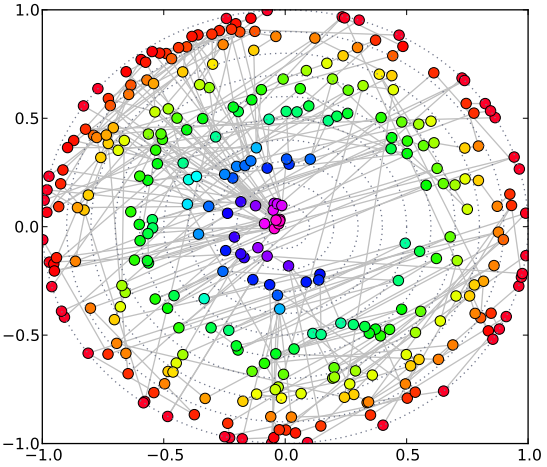


Figure 3.3: Centrality layout of the London tube.

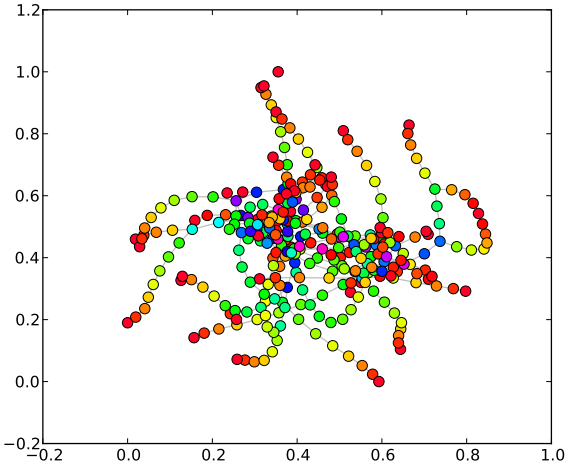


Figure 3.4: Spring layout of the London tube.

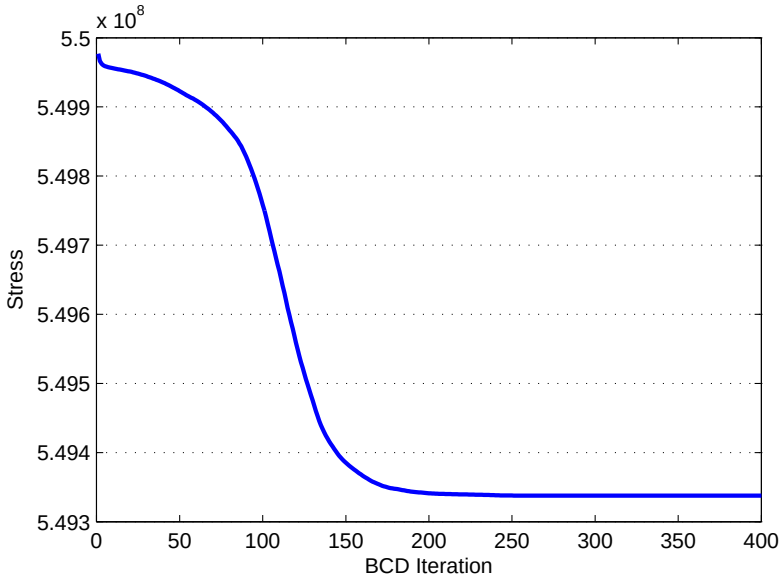
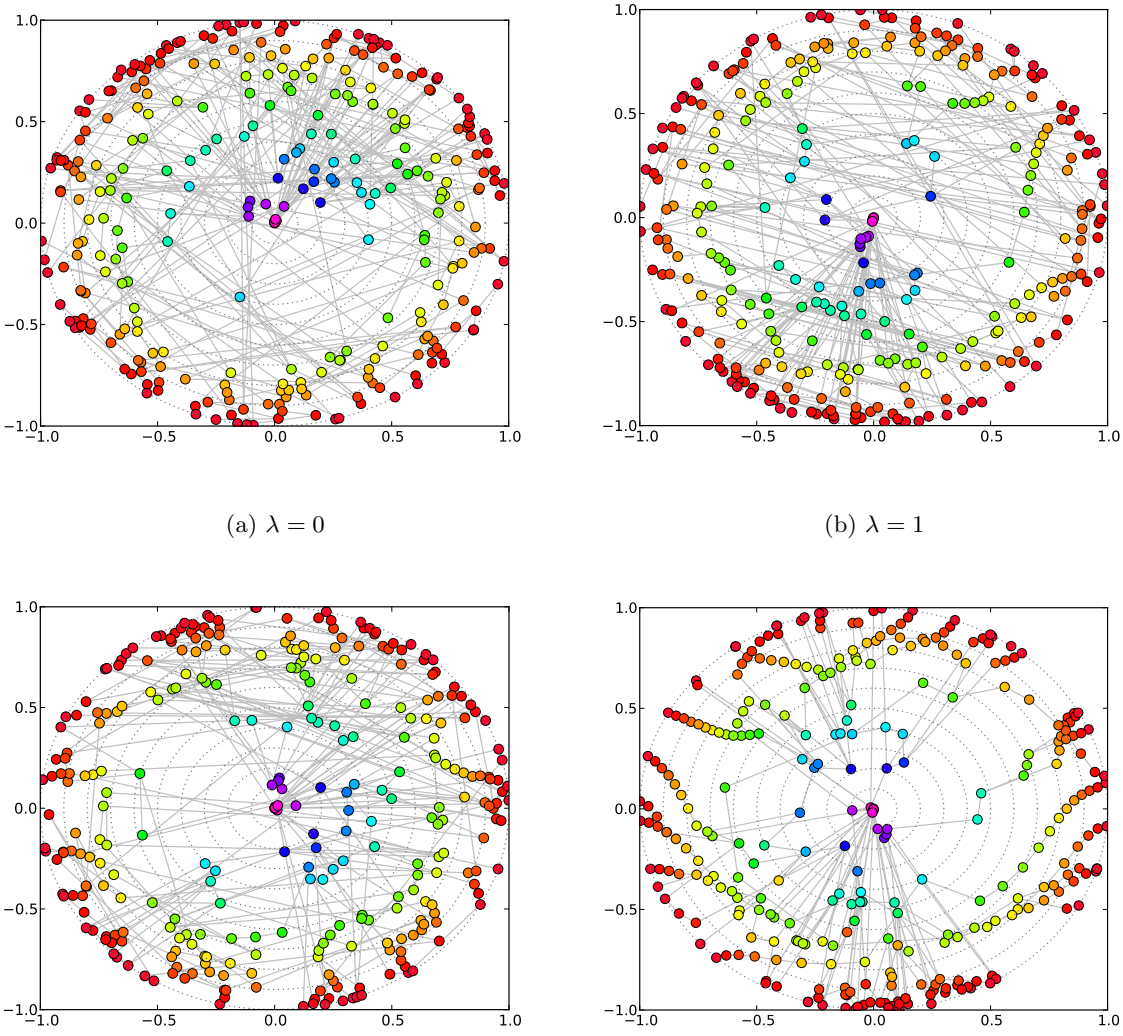


Figure 3.5: MDS stress vs. iterations.



(a) $\lambda = 0$

(b) $\lambda = 1$

(c) $\lambda = 100$

(d) $\lambda = 10,000$

Figure 3.6: Effect of a smoothness penalty: varying λ for the London tube visualization.

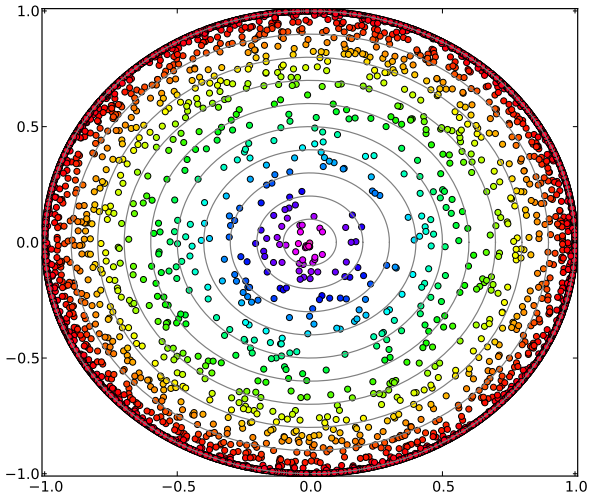


Figure 3.7: MDS embedding of AGR network.

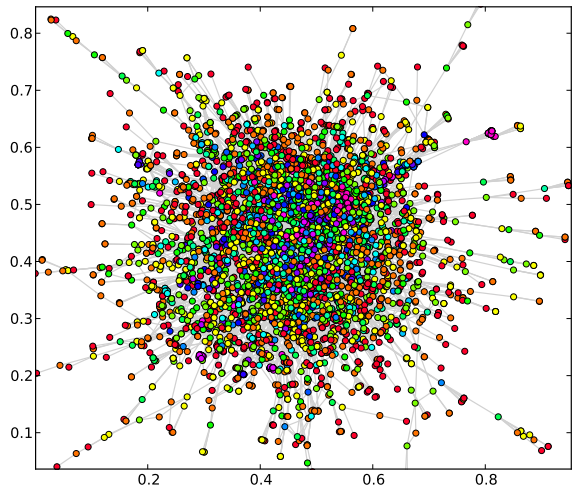


Figure 3.8: Spring layout for AGR network.

Chapter 4

Centrality-constrained LLE

LLE belongs to a family of non-linear dimensionality reduction techniques which assume that the data lie close to a non-linear manifold and seek a local structure preserving embedding in low dimensional Euclidean space. Each high-dimensional point is modeled as a linear combination of its neighbors selected using well-defined criteria. Fitting each point to this linear combination by least-squares yields reconstruction weights that are then preserved by a similar fitting step whose goal entails recovery of a lower-dimensional rendition of the original dataset.

In this chapter, the reconstruction weights are determined under centrality constraints in closed form and the weight preservation step, which turns out to decouple across nodes is solved by coordinate descent iterations. Numerical tests demonstrate that using LLE eliminates the need for smoothness regularization, a result that is not surprising considering its inherent tendency to preserve local structure.

4.1 LLE problem statement

Given data vectors $\{\mathbf{y}_i \in \mathbb{R}^q\}_{i=1}^N$ assumed to lie close to a non-linear manifold in $\mathbb{R}^q$, LLE seeks vectors $\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^N$ where $p \ll q$ by assuming that the manifold is well approximated by some unknown linear subspace within the neighborhood of each datum. The following steps summarize the classical LLE algorithm:

1. Determine the neighborhood $\mathcal{N}_i$ of data point i by selecting its K-nearest neighbors.

2. Fit each point to a linear comination of its neighbors by solving the least-squares problem,

$$\begin{aligned} \underset{w_{i1},\ldots,w_{iK}}{\arg\min} \quad & \|\mathbf{y}_i - \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{y}_j\|_2^2 \\ \text{s. to} \quad & \sum_{j\in\mathcal{N}_i} w_{ij} = 1, \;\; i = 1,\ldots,N \end{aligned} \tag{4.1}$$

   where $\{w_{ij}\}_{j=1}^{K}$ are the K reconstruction weights for point i and the linear constraint enforces shift invariance.

3. Set $w_{ij} = 0$ for $j \notin \mathcal{N}_i$, and determine $\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^{N}$ by least-squares as follows,

$$\begin{aligned} \underset{\mathbf{x}_1,\ldots,\mathbf{x}_N}{\arg\min} \quad & \sum_{i=1}^{N} \|\mathbf{x}_i - \sum_{j=1}^{N} w_{ij}\mathbf{x}_j\|_2^2 \\ \text{s. to} \quad & \sum_{i=1}^{N} \mathbf{x}_i = \mathbf{0}, \quad \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^T = \mathbf{I}. \end{aligned} \tag{4.2}$$

In the final step, the first constraint centers the embedding at the origin whereas the second constraint avoids the trivial solution $\mathbf{x}_i = \mathbf{0}, i = 1,\ldots,N$. The rest of this chapter re-examines steps 2 and 3 and adapts them for graph embedding while adhering to the desired centrality structure.

4.2 Incorporating centrality constraints

Given a graph $\mathcal{G}$, assumed to be undirected, and unweighted with neither self-loops nor multi-edges. In addition, assume all nodes and vertices of $\mathcal{G}$, as well as the vertex centralities $\{c_i\}_{i=1}^{N}$ are known beforehand. Selection of neighbors per node is straightforward and can be accomplished by assigning $\mathcal{N}_i$ to the set of its n-hop neighbors. As a result, each node takes on a different number of neighbors, as dictated by the topology of the graph.

Let node i be assigned a vector, $\mathbf{y}_i$, of arbitrary dimension $q \gg p$, and adding the

centrality constraint $(\sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{y}_j)^T(\sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{y}_j) = f^2(c_i)$ to (4.1) per node yields

$$\begin{aligned} \text{(P4)} \quad \mathbf{w}_i = \underset{\{\mathbf{w}:\, \mathbf{1}^T\mathbf{w}=1\}}{\arg\min} \quad & \|\mathbf{y}_i - \mathbf{Y}_i\mathbf{w}_i\|_2^2 \\ \text{s. to} \quad & \|\mathbf{Y}_i\mathbf{w}_i\|_2^2 = f^2(c_i) \end{aligned} \tag{4.3}$$

where $\mathbf{Y}_i := [\mathbf{y}_1^i, \ldots, \mathbf{y}_K^i]$ contains the K n-hop neighbors of i, $\{\mathbf{y}_j^i\}_{j=1}^K$, and $\mathbf{w}_i := [w_{i1}, \ldots, w_{iK}]^T$. Similarly, the final LLE step is modified as follows

$$\begin{aligned} \text{(P5)} \quad \underset{\mathbf{x}_1,\ldots,\mathbf{x}_N}{\arg\min} \quad & \sum_{i=1}^N \|\mathbf{x}_i - \sum_{j=1}^N w_{ij}\mathbf{x}_j\|_2^2 \\ \text{s. to} \quad & \|\mathbf{x}_i\|_2^2 = f^2(c_i),\ i = 1, \ldots, N \end{aligned} \tag{4.4}$$

with the **0**-mean constraint compensated for by a centering operation after the optimal vectors $\{\mathbf{x}_i\}_{i=1}^N$ have been determined. P4 is nonconvex due to the inclusion of a quadratic equality constraint. Relaxing the constraint to the inequality $(\sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{y}_j)^T(\sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{y}_j) \leq f^2(c_i)$ leads to a convex problem which can be easily solved.

4.3 Solving for reconstruction weights

In general, $\{\mathbf{y}_i\}_{i=1}^N$ are unknown and a graph embedding must be determined entirely from $\mathcal{G}$. However, the terms in both the cost function and the constraint in (4.3) are inner products, $\mathbf{y}_i^T\mathbf{y}_j$ for all $i, j \in \{1, \ldots, N\}$, which can be approximated using the dissimilarities $\{\{\delta_{ij}\}_{j=1}^N\}_{i=j+1}^N$. This is reminiscent of classical MDS which seeks vectors $\{\mathbf{y}_i\}_{i=1}^N$ so that $\|\mathbf{y}_i - \mathbf{y}_j\|_2 \approx \delta_{ij}$ for any pair of points i and j. To this end, (4.3) can be written as

$$\begin{aligned} \mathbf{w}_i = \underset{\{\mathbf{w}:\, \mathbf{1}^T\mathbf{w}=1\}}{\arg\min} \quad & \mathbf{w}^T\mathbf{Y}_i^T\mathbf{Y}_i\mathbf{w} - 2\mathbf{y}_i^T\mathbf{Y}_i\mathbf{w} \\ \text{s. to} \quad & \mathbf{w}^T\mathbf{Y}_i^T\mathbf{Y}_i\mathbf{w} \leq f^2(c_i) \end{aligned} \tag{4.5}$$

If $\boldsymbol{\mathcal{D}}$ denotes the $N \times N$ matrix whose (i, j)th entry is the square Euclidean distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ i.e., $[\boldsymbol{\mathcal{D}}]_{ij} := \|\mathbf{y}_i - \mathbf{y}_j\|_2^2$, and $\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, then

$$\boldsymbol{\mathcal{D}} = \boldsymbol{\psi}\mathbf{1}^T + \mathbf{1}\boldsymbol{\psi}^T - 2\mathbf{Y}^T\mathbf{Y} \tag{4.6}$$

where $\boldsymbol{\psi} \in \mathbb{R}^N$ has entries $\psi_i = \|\mathbf{y}_i\|_2^2$. Denoting the centering matrix as $\mathbf{J} := \mathbf{I} - N^{-1}\mathbf{1}\mathbf{1}^T$, and applying the double-centering operation to $\boldsymbol{\mathcal{D}}$ yields

$$\begin{aligned} -\frac{1}{2}\mathbf{J}\boldsymbol{\mathcal{D}}\mathbf{J} &= -\frac{1}{2}\mathbf{J}\boldsymbol{\psi}\mathbf{1}^T\mathbf{J} - \frac{1}{2}\mathbf{J}\mathbf{1}\boldsymbol{\psi}^T\mathbf{J} + \mathbf{J}\mathbf{Y}^T\mathbf{Y}\mathbf{J} \\ &= \mathbf{Y}^T\mathbf{Y} \end{aligned} \tag{4.7}$$

which is the inner product matrix of the vectors $\{\mathbf{y}_i\}_{i=1}^N$, i.e., $[\mathbf{Y}^T\mathbf{Y}]_{ij} = \mathbf{y}_i^T\mathbf{y}_j$, since $\mathbf{Y}^T\mathbf{Y}$ is already double-centered. Although $\boldsymbol{\mathcal{D}}$ is unknown, it is possible to estimate $\mathbf{Y}^T\mathbf{Y} \approx -\frac{1}{2}\mathbf{J}\boldsymbol{\Delta}^{(2)}\mathbf{J}$ from (4.7) using square dissimilarity measurements $[\boldsymbol{\Delta}^{(2)}]_{ij} = \delta_{ij}^2$ as surrogates to the square Euclidean distances. Since $\boldsymbol{\Delta}^{(2)}$ can be determined from the graph topology alone using graph-theoretic distance measures (e.g., shortest path distances), (4.5) can be determined in closed form as shown next. If $\mathbf{H} := -\frac{1}{2}\mathbf{J}\boldsymbol{\Delta}^{(2)}\mathbf{J}$, and $\mathbf{H}_i$ denotes the estimate of $\mathbf{Y}_i^T\mathbf{Y}_i$, then $\mathbf{H}_i = \mathbf{H}_{\mathcal{N}_i,\mathcal{N}_i}$. Similarly, the estimate of $\mathbf{Y}_i^T\mathbf{y}_i$ is $\mathbf{h}_i = \mathbf{H}_{\mathcal{N}_i,i}$. Using this notation, the first step of LLE in the proposed approach seeks the solution to the convex constrained QP

$$\begin{aligned} \text{(P6)}\quad \mathbf{w}_i = \underset{\{\mathbf{w}:\, \mathbf{1}^T\mathbf{w}=1\}}{\arg\min} \quad & \mathbf{w}^T\mathbf{H}_i\mathbf{w} - 2\mathbf{h}_i^T\mathbf{w} \\ \text{s. to} \quad & \mathbf{w}^T\mathbf{H}_i\mathbf{w} \leq f^2(c_i), i = 1, \ldots, N. \end{aligned} \tag{4.8}$$

Remark 4.1. *Since the entries of* $\mathbf{H}$ *are inner products, it is a kernel matrix and can be replaced by a graph kernel that reliably captures similarities between nodes in* $\mathcal{G}$*. The Laplacian pseudoinverse,* $\mathbf{L}^\dagger$ *is such a kernel in the space spanned by the graph nodes, where node* i *is represented by* $\mathbf{e}_i$ *[40]. In fact,* $(-1/2)\mathbf{J}\boldsymbol{\Delta}^{(2)}\mathbf{J}$ *and* $\mathbf{L}^\dagger$ *are equivalent when the entries of* $\boldsymbol{\Delta}^{(2)}$ *are average commute times, briefly discussed in Section 2.3.*

In order to solve P6 per node, Lagrange multipliers γ and μ corresponding to the inequality and equality constraints are introduced and lead to the Lagrangian

$$L(\mathbf{w}, \gamma, \mu) = \mathbf{w}^T\mathbf{H}_i\mathbf{w} - 2\mathbf{h}_i^T\mathbf{w} - \gamma(\mathbf{w}^T\mathbf{H}_i\mathbf{w} - f^2(c_i)) + \mu(\mathbf{1}^T\mathbf{w} - 1). \tag{4.9}$$

Assuming Slater's condition is satisfied [44], a zero duality gap is achieved if the following KKT conditions are satisfied by the optimal primal and dual variables:

$$\begin{aligned} \mathbf{w}^T\mathbf{H}_i\mathbf{w} - f^2(c_i) &\leq 0 && \text{(4.10a)} \\ \mathbf{1}^T\mathbf{w} - 1 &= 0 && \text{(4.10b)} \\ \gamma &\geq 0 && \text{(4.10c)} \\ \gamma(\mathbf{w}^T\mathbf{H}_i\mathbf{w} - f^2(c_i)) &= 0 && \text{(4.10d)} \\ \nabla L(\mathbf{w},\gamma,\mu) = (1+\gamma)\mathbf{H}_i\mathbf{w} + \frac{\mu}{2}\mathbf{1} - \mathbf{h}_i &= 0. && \text{(4.10e)} \end{aligned}$$

Upon solving the KKT conditions (see Appendix B for derivation) it follows that

$$\mu^* = \frac{2(\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i)}{\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1}} + 2\left\{\frac{\mathbf{h}_i^T\mathbf{H}_i^{-1}(\mathbf{1}\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i - \mathbf{h}_i\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1})}{(\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1})^2 - (\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1})^3 f^2(c_i)}\right\}^{\frac{1}{2}} \tag{4.11}$$

and

$$\gamma^* = \mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i - \frac{\mu^*}{2}\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1} - 1. \tag{4.12}$$

The optimal value, $\mathbf{w}_i^*$, is determined as follows

$$\mathbf{w}_i^* = \begin{cases} \frac{1}{(1+\gamma^*)}\mathbf{H}_i^{-1}\left(\mathbf{h}_i - \frac{\mu^*}{2}\mathbf{1}\right), & \text{if } \gamma^* \in \mathbb{R}_+ \\ \mathbf{H}_i^{-1}\left(\mathbf{h}_i - \frac{\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i - 1}{\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1}}\mathbf{1}\right), & \text{otherwise.} \end{cases} \tag{4.13}$$

Although $\mathbf{H}_i$ is provably positive semidefinite, it is not guaranteed to be non-singular as required by (4.11) - (4.13). Since $\mathbf{H}_i$ is an approximation to inner products, ensuring positive definiteness by uniformly perturbing the diagonal entries slightly is reasonable in this case, i.e., $\mathbf{H}_i \leftarrow \mathbf{H}_i + \sigma\mathbf{I}_K$, where σ is a small number.

4.4 Determination of embedding vectors

Setting all $w_{ij} = 0$ for $j \notin \mathcal{N}_i$, the sought graph embedding is determined by solving P5 over $\{\mathbf{x}_i\}_{i=1}^N$. P5 is non-convex and global optimality is not guaranteed. However, the problem decouples over vectors $\{\mathbf{x}_i\}_{i=1}^N$ and this block separability can be exploited to solve it via BCD iterations. Inner iteration i under outer iteration r involves solving

$$\begin{aligned} \mathbf{x}_i^r = \arg\min_{\mathbf{x}} \quad & \|\mathbf{x} - \sum_{j<r} w_{ij}\mathbf{x}_j^r - \sum_{j>r} w_{ij}\mathbf{x}_j^{r-1}\|_2^2 \\ \text{s. to} \quad & \|\mathbf{x}\|_2^2 = f^2(c_i). \end{aligned} \tag{4.14}$$

With $\mathbf{v}_i^r := \sum_{j<r} w_{ij}\mathbf{x}_j^r + \sum_{j>r} w_{ij}\mathbf{x}_j^{r-1}$ and ν denoting a Lagrange multiplier, minimization of the Lagrangian for (4.14) leads to

$$\mathbf{x}_i^r = \arg\min_{\mathbf{x}} \quad \|\mathbf{x} - \mathbf{v}_i^r\|_2^2 + \nu(\|\mathbf{x}\|_2^2 - f^2(c_i)) \tag{4.15}$$

which yields

$$\mathbf{x}_i^r = \frac{\mathbf{v}_i^r}{1+\nu}. \tag{4.16}$$

Substituting (4.16) into the equality constraint in (4.14) leads to the closed-form per-iteration update of $\mathbf{x}_i$

$$\mathbf{x}_i^r = \begin{cases} \frac{\mathbf{v}_i^r}{\|\mathbf{v}_i^r\|_2} f(c_i), & \text{if } \|\mathbf{v}_i^r\|_2 > 0 \\ \mathbf{x}_i^{r-1}, & \text{otherwise.} \end{cases} \tag{4.17}$$

Letting $\mathbf{X}^r$ denote the embedding matrix after r BCD iterations, the operation $\mathbf{X} = (\mathbf{I} - N^{-1}\mathbf{1}\mathbf{1}^T)\mathbf{X}^r$ centers $\{\mathbf{x}_i^r\}_{i=1}^N$ to the origin in order to satisfy the shift invariance property of the embedding.

Algorithm 3 summarizes the steps outlined in this section for the LLE-based graph embedding approach. It is assumed that the only inputs to the algorithm are the graph topology $\mathcal{G}$, the centrality measures, $\{c_i\}_{i=1}^N$, the graph embedding dimension p, the square dissimilarity matrix $\mathbf{\Delta}^{(2)}$, and the number of hops to consider for neighborhood selection per node, n.

4.5 Numerical experiments

In this section, the proposed approaches are used to visualize two real-world networks that serve entirely different purposes. Appropriate centrality measures are selected to highlight particular aspects of the two networks.

Algorithm 3 Graph embedding via LLE

Input: $\mathcal{G}$, $\{c_i\}_{i=1}^N$, $\mathbf{\Delta}^{(2)}$, ϵ, n, p
Set $\mathbf{H} = -\frac{1}{2}\mathbf{J}\mathbf{\Delta}^{(2)}\mathbf{J}$
for $i = 1 \dots N$ **do**
  Set $\mathcal{N}_i$ to n-hop neighbors of i
  $\mathbf{H}_i = \mathbf{H}_{\mathcal{N}_i,\mathcal{N}_i}$, $\mathbf{h}_i = \mathbf{H}_{\mathcal{N}_i,i}$
  Compute μ, γ from (4.11), (4.12)
  Compute $\mathbf{w}_i$ from (4.13)
  Set $w_{ij} = 0$ for $j \notin \mathcal{N}_i$
end for
Initialize $\mathbf{X}^0$, $r = 0$
repeat
  $r = r + 1$
  for $i = 1 \dots N$ **do**
    Compute $\mathbf{x}_i^r$ according to (4.17)
    $\mathbf{X}^r(i,:) = (\mathbf{x}_i^r)^T$
  end for
until $\|\mathbf{X}^r - \mathbf{X}^{r-1}\|_F \leq \epsilon$
$\mathbf{X} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)\mathbf{X}^r$

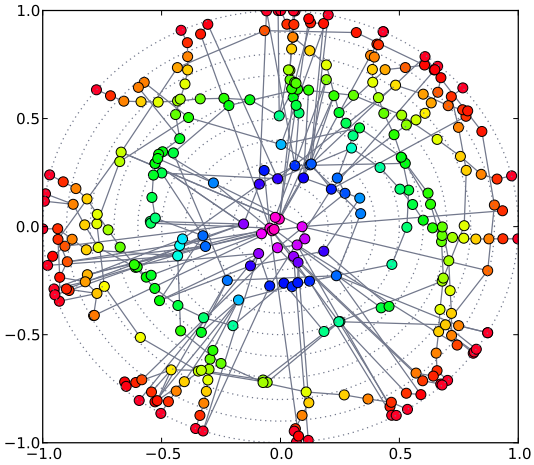


Figure 4.1: London tube.

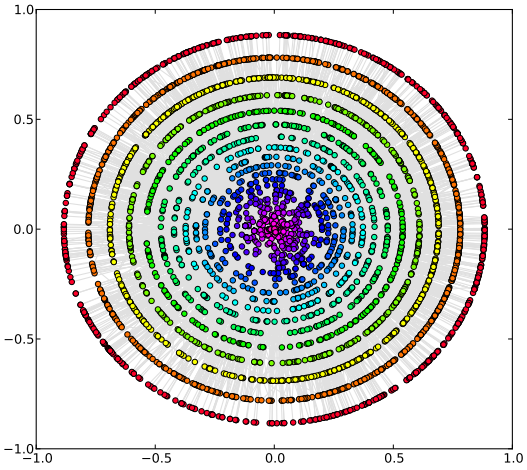


Figure 4.2: AGR network.

4.5.1 The London tube

Using the same centrality transformation, Algorithm 3 was run with $n = 1$ (single-hop neighbors). Figure 4.1 depicts the resulting embedding after 20 BCD iterations in the final LLE step. For the initialization of the BCD iterations, $\mathbf{X}^0$ was generated from a Gaussian distribution of unit covariance matrix, i.e., $\mathbf{x}_i^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for all $i = 1, \ldots, N$. Visual inspection of the embeddings from the two approaches reveals that using LLE naturally minimizes the number of edge-crossings. This behavior is a direct result of the implicit constraint that the embedding vector assigned to each node is only influenced by the immediate neighbors. However, this benefit comes at the cost of the lack of a convergence guarantee for the BCD iterations.

4.5.2 AGR network

The LLE algorithm was run for the same data yielding the network visualization depicted by Figure 4.2. Only single-hop neighbors were considered ($n = 1$) for the neighborhood selection. Although no claims of optimal placement or convergence are made for Algorithm 3, it is clear that the visualization succeeds at conveying the centrality structure of the

collaboration network. By comparison, Figure 3.8 depicts the same network drawn via a springer-embedder model, highlighting the fact that little can be said of the structural properties of the network.

Chapter 5

Conclusion

5.1 Summary

In this thesis, two approaches for embedding graphs with certain structural constraints were proposed. In the first approach, an optimization problem was formulated under centrality constraints that capture relative levels of importance between nodes. A block coordinate descent solver with successive approximations was developed to deal with the non-convexity and non-smoothness of the constrained MDS stress minimization problem. In addition, a smoothness penalty term was incorporated to minimize edge crossings in the resultant network visualizations. Tests on real-world networks were run and the results demonstrated that convergence is guaranteed, and large networks can be visualized relatively fast.

In the second proposed approach, LLE was adapted to the visualization problem by solving centrality-constrained optimization problems for both steps of the algorithm. The first step, which determines reconstruction weights per node, amounted to a QCQP that was solved in closed-form. The final step which determines the embedding that best preserves the weights turned out to decouple across nodes and was solved via BCD iterations with closed-form solutions per subproblem. Despite the lack of an optimality guarantee for this step, meaningful network visualizations were obtained after a few iterations making the developed algorithm attractive for large scale graph embeddings.

From an application perspective, the LLE approach is preferred over MDS in settings

where the network represents data sampled from an underlying manifold. For instance, consider a social network whose nodes represent people and edges encode self-reported friendship ties between them. It is assumed that friends have a number of similar attributes e.g., closeness in age, level of education, and income bracket. Although some of these attributes may be unknown, it is reasonable to assume that they lie near a low-dimensional nonlinear manifold, nicely motivating an LLE approach to graph embedding.

Some real-world networks are heterogeneous with multi-typed nodes and edges e.g., a bibliographic network that captures the relationships between authors, conferences and papers. The smooth assumption does not apply here, but dissimilarity measures between multi-typed nodes have been developed [51], and an MDS approach is well justified.

5.2 Future directions

The developed network visualization approaches hinge upon several assumptions, namely: i) networks of interest are static and do not change significantly over the observation period, and ii) complete knowledge of the constituent nodes and edges. Although these assumptions are reasonable for some networks e.g., transportation networks, many complex systems are described by large, dynamic networks rendering the acquisition of complete snapshots of their topology at any given time point considerably challenging. To this end, the following axes have been identified for future related research efforts:

1. Development of visualization approaches that efficiently deal with limited centrality and node dissimilarity data due to network sampling.

2. Capturing temporal variations of the structural properties of large dynamic networks such as the physical Internet infrastructure and online social networks.

3. In certain networks, nodes are endowed with known attributes regardless of the underlying topology. Network visualization approaches that capture both topological structure and patterns in the node attributes are well motivated in this setting.

Bibliography

[1] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, Springer, New York, NY, USA, 2009.

[2] M. E. J. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, pp. 167–256, June 2003.

[3] B. Luo, R. C. Wilson, and E. R. Hancock, "Spectral embedding of graphs," *Pattern Recognition*, vol. 36, pp. 2213–2230, January 2003.

[4] B. Shaw and T. Jebara, "Structure preserving embedding," *International Conference of Machine Learning*, Montreal, Canada, June 2009.

[5] C. D. Correa, T. Crnovrsanin, and K.-L. Ma, "Visual reasoning about social networks using centrality sensitivity," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 106–120, January 2012.

[6] U. Dogrusoz, M. E. Belviranli, and A. Dilek, "CiSE: A circular spring embedder layout algorithm," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 953–966, June 2013.

[7] J. Yang, Y. Liu, X. Zhang, X. Yuan, Y. Zhao, S. Barlowe, and S. Liu, "PIWI: Visually exploring graphs based on their community structure," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 1034–1047, June 2013.

[8] J. Abello, F. V. Ham, and N. Krishnan, "ASK-GraphView : A large scale graph visualization system," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 669–676, September 2006.

[9] M. Girvan and M. E.J. Newman, "Community structure in social and biological networks," *Proc. of the National Academy of Sciences U.S.A.*, vol. 99, pp. 7821–7826, April 2002.

[10] Y. Jia, J. Hoberock, M. Garland, and J. Hart, "On the visualization of social and other scale-free networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1285–1292, October 2008.

[11] F. V. Ham and M. Wattenberg, "Centrality based visualization of small world graphs," *Computer Graphics Forum*, vol. 27, pp. 975–982, May 2008.

[12] E. Giacomo, W. Didimo, L. Grilli, and G. Liotta, "Graph visualization techniques for web clustering engines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 294–304, March 2007.

[13] L. Harrison and A. Lu, "The future of security visualization: Lessons from network visualization," IEEE Network, vol. 26, pp. 6–11, December 2012.

[14] J. Ignatowicz, "Drawing force-directed graphs using optigraph," *Proceedings of DIMACS International Workshop*, vol. 894, pp. 333–336, October 1994.

[15] A. Papakostas and I. Tollis, "Algorithms for area-efficient orthogonal drawings," *Computational Geometry: Theory and Applications*, vol. 9, pp. 83–110, January 1998.

[16] E. Reingold and J. Tilford, "Tidier drawing of trees," *IEEE Trans. on Software Engineering*, vol. 7, pp. 223–228, March 1981.

[17] K. Supowit and E. Reingold, "The complexity of drawing trees nicely," *Acta Informatica*, vol. 18, pp. 359–368, January 1983.

[18] C. Wetherell and A. Shannon, "Tidy drawing of trees," *IEEE Trans. on Software Engineering*, vol. 5, pp. 514–520, September 1979.

[19] T. Biedl and G. Kant, "A better heuristic for orthogonal graph drawings," *Proceedings of the Second Annual European Symposium on Algorithms*, vol. 855, pp. 24–35, September 1994.

[20] D. Tunkelang, "A numerical optimization approach to general graph drawing," *PhD Thesis, Carnegie Mellon University School of Computer Science*, January 1999.

[21] T. Fruchterman, and E. Reingold, "Graph drawing by force-directed placement," *Software Practice and Experience*, vol. 21, pp. 1129-1164, November 1991.

[22] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, pp. 581-683, December 1966.

[23] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, pp. 35–41, March 1977.

[24] P. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et des Jura," *Bulletin de la Societé Vaudoise des Sceience Naturelles*, vol. 37, pp. 547–579, November 1901.

[25] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, A. L. Barabási, "Hierarchical organization of modularity in metabolic networks", *Science*, vol. 297, pp. 1551–1555, August 2002.

[26] E. A. Leicht, P. Holme, M. E. J. Newman, "Vertex similarity in networks", *Physical Review E*, vol. 73, 026120, February 2006.

[27] D. J. Klein, M. Randic, "Resistance distance", *Journal of Mathematical Chemistry*, vol. 12, pp. 81–95, December 1993.

[28] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: bringing order to the web.", *Stanford InfoLab*, November 1999.

[29] W. Liu, L. Lü, Jin, T. Zhou, "Link prediction based on local random walk," *Europhysics Letters*, vol. 89, 58007, March 2010.

[30] U. Brandes and C. Pich, "More flexible radial layout," *Journal of Graph Algorithms and Applications*, vol. 15, pp. 157–173, February 2011.

[31] I. Borg and P. J. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer, New York, NY, 2005.

[32] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," em Neural computation, vol. 15, pp. 1373–1396, June 2003.

[33] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, pp. 7–15, April 1989.

[34] L. K. Saul, and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research*, vol. 4, pp. 119–155, June 2003.

[35] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, December 2000.

[36] D. De Ridder, O. Kouropteva, O. Okun, M. Pietikainen, and R. Duin, "Supervised locally linear embedding," *Artificial Neural Networks and Neural Information Processing-ICANN/ICONIP*, pp. 175–175, June 2003.

[37] H. Chang and Y. Dit-Yan, "Robust locally linear embedding," *Pattern Recognition*, vol. 39, pp. 1053–1065, June 2006.

[38] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, USA, 2001.

[39] J. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani, "Large scale networks fingerprinting and visualization using the k-core decomposition," *Advances in Neural Information Processing Systems*, vol. 18, pp. 41–50, May 2006.

[40] F. Fouss, A. Pirotte, J. M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 355–369, March 2007.

[41] S. L. France and J. D. Carroll, "Two-way multidimensional scaling: a review," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 41, pp. 644–661, September 2011.

[42] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 988–995, July 2004.

[43] K. Q. Weinberger, B. D. Packer, and L. K. Saul, "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization," *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pp. 381–388, January 2005.

[44] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, 2004.

[45] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, Belmont, MA, 1999.

[46] M. Razaviyayn, M. Hong, and Z. Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *arXiv preprint arXiv:1209.2385v1*, September 2012.

[47] B. Baingana, and G. B. Giannakis, "Centrality-constrained graph embedding," *Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, Vancouver, Canada, May 26 - 31, 2013.

[48] K. Q. Weinberger, F. Sha, Q. Zhu, and L. K. Saul, "Graph Laplacian regularization for large-scale semidefinite programming," *Advances in Neural Information Processing Systems*, vol. 19, pp. 1489–1496, September 2007.

[49] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, "Data visualization with multidimensional scaling," *Journal of Comp. and Graph. Stats.*, pp. 444–472, June 2008.

[50] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, article 2, March 2007.

[51] C. Shi, X. Kong, P. S. Yu, S. Xie, and B. Wu, "Relevance search in heterogeneous networks," *Proceedings of the 15th International Conf. on Extending Database Technology*, pp. 180–191, March 2012.

Appendices

Appendix A

Proof of proposition 3.1.

Substituting $\mathbf{x}_0 = \mathbf{x}^{r-1}$ in the LHS of (3.12a) yields

$$\begin{aligned}\Phi(\mathbf{x}^{r-1}, \mathbf{x}^{r-1}) &= \psi_1(\mathbf{x}^{r-1}) - \psi_2'(\mathbf{x}^{r-1}, \mathbf{x}^{r-1}) \\ &= \frac{(N-1)}{2}\|\mathbf{x}^{r-1}\|_2^2 - (\mathbf{x}^{r-1})^T(\sum_{j<i}\mathbf{x}_j^r + \sum_{j>i}\mathbf{x}_j^{r-1}) \\ &\quad - \sum_{j<i}\delta_{ij}\|\mathbf{x}^{r-1} - \mathbf{x}_j^r\|_2 - \sum_{j>i}\delta_{ij}\|\mathbf{x}^{r-1} - \mathbf{x}_j^{r-1}\|_2. \qquad \text{(A.1)}\end{aligned}$$

Since

$$\begin{aligned}\Psi(\mathbf{x}^{r-1}) &= \psi_1(\mathbf{x}^{r-1}) - \psi_2(\mathbf{x}^{r-1}) \\ &= \frac{(N-1)}{2}\|\mathbf{x}^{r-1}\|_2^2 - (\mathbf{x}^{r-1})^T(\sum_{j<i}\mathbf{x}_j^r + \sum_{j>i}\mathbf{x}_j^{r-1}) \\ &\quad - \sum_{j<i}\delta_{ij}\|\mathbf{x}^{r-1} - \mathbf{x}_j^r\|_2 - \sum_{j>i}\delta_{ij}\|\mathbf{x}^{r-1} - \mathbf{x}_j^{r-1}\|_2, \qquad \text{(A.2)}\end{aligned}$$

condition (3.12a) is satisfied by the equivalence of (A.1) and (A.2). The second condition, (3.12b), is automatically satisfied for $\mathbf{x}_0 = \mathbf{x}^{r-1}$ by (3.7).

Appendix B

Solving for the KKT conditions in section 4.3

This appendix derives (4.10a) - (4.10e) from the KKT conditions in (4.11) - (4.13). From (4.10e), one obtains

$$2\mathbf{H}_i\mathbf{w}_i^* - 2\mathbf{h}_i + 2\gamma^*\mathbf{H}_i\mathbf{w}_i^* + \mu^*\mathbf{1} = \mathbf{0}. \tag{B.1}$$

Solving for $\mathbf{w}_i^*$ in terms of γ^* and μ^* from (B.1)

$$\mathbf{w}_i^* = \frac{1}{(1+\gamma^*)}\mathbf{H}_i^{-1}\left(\mathbf{h}_i - \frac{\mu^*}{2}\mathbf{1}\right). \tag{B.2}$$

Upon applying the result in (B.2) to the primal feasibility condition in (4.10b), it turns out that

$$\mathbf{1}^T\left[\frac{\mathbf{H}_i^{-1}}{(1+\mu^*)}\left(\mathbf{h}_i - \frac{\mu^*}{2}\mathbf{1}\right)\right] = 1. \tag{B.3}$$

Thus,

$$\gamma^* = \mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i - \frac{\mu^*}{2}\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1} - 1. \tag{B.4}$$

Assuming the inequality constraint is inactive at optimality, then $\gamma^* = 0$, which upon solving (B.4) yields

$$\mu^* = \frac{2\left(\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i - 1\right)}{\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1}}. \tag{B.5}$$

Combining (B.5) and (B.2) gives

$$\mathbf{w}_i^* = \mathbf{H}_i^{-1}\left[\mathbf{h}_i - \frac{\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i - 1}{\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1}}\mathbf{1}\right]. \tag{B.6}$$

If the inequality constraint is active at optimality, then

$$\mathbf{w}_i^{*T}\mathbf{H}_i\mathbf{w}_i^* = f^2(c_i). \tag{B.7}$$

Substituting for $\mathbf{w}_i^*$ from (B.2) results in

$$(2\mathbf{h}_i - \mu^*\mathbf{1})^T\,\mathbf{H}_i^{-1}\,(2\mathbf{h}_i - \mu^*\mathbf{1}) = 4f^2(c_i)(1+\gamma^*)^2 \tag{B.8}$$

which upon simplification yields

$$\mathbf{h}_i^T\mathbf{H}_i^{-1}\mathbf{h}_i - \mu^*\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i + \frac{\mu^{*2}}{4}\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1} = f^2(c_i)(1+\gamma^*)^2. \tag{B.9}$$

Substituting for γ^* results in the following quadratic equation in μ^*

$$\mathbf{h}_i^T\mathbf{H}_i^{-1}\mathbf{h}_i - \mu^*\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i + \frac{\mu^{*2}}{4}\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1} = f^2(c_i)\left(\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i - \frac{\mu^*}{2}\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1}\right)^2. \tag{B.10}$$

Finally, solving for μ^* from (B.10) and taking one root yields

$$\mu^* = 2\frac{\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i}{\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1}} + 2\left\{\frac{\mathbf{h}_i^T\mathbf{H}_i^{-1}(\mathbf{1}\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{h}_i - \mathbf{h}_i\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1})}{(\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1})^2 - (\mathbf{1}^T\mathbf{H}_i^{-1}\mathbf{1})^3 f^2(c_i)}\right\}^{\frac{1}{2}}. \tag{B.11}$$