KERNEL-BASED EMBEDDINGS FOR LARGE GRAPHS WITH CENTRALITY CONSTRAINTS

Brian Baingana and Georgios B. Giannakis

Dept. of ECE and Digital Technology Center, University of Minnesota, Minneapolis, MN

ABSTRACT

Complex phenomena involving pairwise interactions in natural and man-made settings can be well-represented by networks. Besides statistical and computational analyses on such networks, visualization plays a crucial role towards effectively conveying "at-a-glance" structural properties such as node hierarchy. However, most graph embedding algorithms developed for network visualization are ill-equipped to cope with the sheer volume of data generated by modern networks that encompass online social interactions, the Internet, or the world-wide web. Motivated by the emergence of nonlinear manifold learning approaches for dimensionality reduction, this paper puts forth a novel scheme for embedding graphs using kernel matrices defined on graphs. In particular, a kernelized version of local linear embedding is devised for computation of reconstruction weights. Unlike contemporary approaches, the developed embedding algorithm entails low-cost, parallelizable, and closed-form updates that can easily scale to big network data. Furthermore, it turns out that inclusion of embedding constraints to emphasize centrality structure can be accomplished at minimal extra computational cost. Experimental results on Watts-Strogatz small-world networks demonstrate the efficacy of the novel approach.

Index Terms— Graph embedding, local linear embedding, network visualization, coordinate descent.

1. INTRODUCTION

Complex networks abound due to phenomena as diverse as online social interactions, financial transactions among banks, and the Internet [20]. Visual analysis of such networks is fundamentally challenging due to the sheer volume of data often involved. Network visualization is an application of graph embedding which maps each node to a point in Euclidean space. The rising complexity of networks presents new challenges and opportunities for graph embedding tools that effectively capture global patterns and convey meaningful structural information such as hierarchy, similarity and natural communities [14].

Traditional visualization algorithms on small graphs tend to emphasize aesthetics like minimal edge crossing over underlying structural patterns, see e.g., [8, 17, 21]. As a result, they are ill-equipped to visualize larger graphs. Nevertheless, several approaches have been developed for embedding graphs while preserving specific structural properties. Pioneering methods (e.g., [17]) leverage multidimensional scaling (MDS), which seeks a low-dimensional representation of data in which pairwise distances between embedding coordinates are as close to the dissimilarities between the original data points as possible [4].

Work in this paper was supported by the NSF ECCS Grant No. 1343248, and the NSF AST Grant No. 1247885.

Spectral embedding which amounts to principal component analysis (PCA) on the rows of the graph adjacency matrix is advocated in [23]. The structure-preserving embedding algorithm [25] solves a semidefinite program with linear topology constraints so that a nearest neighbor algorithm can recover the graph edges from the embedding. Certain visual analytics approaches emphasize community structure with applications to community browsing in graphs [28]. Concentric graph layouts developed in [1, 2, 5] capture notions of node hierarchy by placing the highest ranked nodes at the center of the embedding.

Since graph embedding boils down to a dimensionality reduction task, it is no surprise that "workhorse" approaches (e.g., PCA and MDS) have risen to prominence in network visualization. However, manifold learning approaches have not received sufficient attention for embedding graphs. In order to effectively preserve the local graph topology structure, the present paper advocates a novel approach that leverages local linear embedding (LLE) [24]. Unlike traditional LLE which presupposes the availability of high dimensional vectors, the novel approach permeates benefits from kernel-based methods to compute the reconstruction weights per node. In addition, the hidden cost of selecting neighbors per datum is easily circumvented by resorting to single-hop neighbors per node.

The growing interest in analysis of large networks has prioritized the need to effectively capture hierarchy over aesthetic appeal in visualization. For instance, a hierarchy-aware visual analysis of a global computer network is more useful to security analysts trying to protect the most critical nodes from a virus attack. Layouts of metro-transit networks that clearly show terminals routing the bulk of traffic convey an informative picture about the most critical nodes in the event of a terrorist attack. This paper incorporates centrality constraints in order to capture node hierarchy. Centralities are a family of well-defined measures of node importance that can facilitate hierarchical ordering of graph nodes; see e.g., [10, 22]. The developed algorithm is simple, incurs a low computational overhead, outperforms spectral approaches in running time, and effectively captures both local and hierarchical structure.

The rest of the paper is organized as follows. Section 2 formally states the problem and Section 3 focuses on the novel kernel-based approach. Centrality constraints are introduced in Section 4, experimental results are presented in Section 5, and concluding remarks are given in Section 6.

2. PROBLEM STATEMENT

Consider a network represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ denotes the set of edges, and $\mathcal{V}$ the set of vertices with cardinality $|\mathcal{V}| = N$. Suppose the structure of $\mathcal{G}$ is captured by an adjacency matrix $\mathbf{A}$ whose (i, j)-th entry (hereafter denoted by a_{ij}) is zero only if edge $(i, j) \notin \mathcal{E}$, otherwise it denotes the weight of (i, j). Given $\mathcal{G}$ and a prescribed embedding dimension p (typically

$p \in \{2, 3\}$), the graph embedding task is tantamount to searching for the set of $p \times 1$ vectors $\mathcal{X} := \{\mathbf{x}_i\}_{i=1}^N$ which "effectively" capture the underlying local graph structure.

This paper advocates LLE, a non-linear dimensionality reduction technique with well-documented merits for preserving local structure by assuming that high-dimensional input data are sampled from a low-dimensional manifold [15, Chap. 14]. LLE accomplishes this by approximating each datum by a linear combination of its neighbors determined by a well-defined criterion followed by construction of a lower dimensional embedding that best preserves the approximations. The next section adapts this manifold learning approach to the graph embedding problem.

3. KERNEL-BASED GRAPH EMBEDDING

Prior to a formal description of the novel graph embedding approach, it is necessary to give a brief outline of LLE. Suppose $\{\mathbf{y}_i \in \mathbb{R}^q\}_{i=1}^N$ are data points sampled from a nonlinear manifold within high dimensional space $\mathbb{R}^q$. LLE seeks the low-dimensional vectors $\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^N$ ($p \ll q$) that preserve the local neighborhood structure on the manifold. To accomplish this, one first constructs the neighborhoods $\{\mathcal{N}_i\}_{i=1}^N$ indexed per datum. Typically, $\mathcal{N}_i$ is set to the K-nearest neighbors of i, or $\mathcal{N}_i := \{\mathbf{y}_j \in \mathbb{R}^q : \|\mathbf{y}_i - \mathbf{y}_j\|_2 \leq \epsilon, \epsilon > 0, j = 1, \ldots, N\}$. Without loss of generality, assuming $|\mathcal{N}_i| = K$, each point is then fit to a linear combination of its neighbors by solving the following constrained least-squares (LS) optimization problem

$$\underset{\left\{\sum_{j\in\mathcal{N}_i} w_{ij}=1\right\}}{\arg\min} \left\| \mathbf{y}_i - \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{y}_j \right\|_2^2 \quad i = 1, \ldots, N \tag{1}$$

where $\{w_{ij}\}_{j=1}^K$ are the K reconstruction weights for point i and the constraint enforces shift invariance. Setting $w_{ij} = 0$ for $j \notin \mathcal{N}_i$, the final step determines $\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^N$ that preserve neighborhood structure through the reconstruction weights by solving

$$\underset{\left\{\begin{smallmatrix}\mathbf{x}_1,\ldots,\mathbf{x}_N \\ \sum_{i=1}^N \mathbf{x}_i=\mathbf{0} \\ \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i\mathbf{x}_i^\top=\mathbf{I}\end{smallmatrix}\right\}}{\arg\min} \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{j=1}^N w_{ij}\mathbf{x}_j \right\|_2^2. \tag{2}$$

The equality constraints center the embedding at the origin with unit sample covariance, thus avoiding the trivial all-zero solution.

For the general graph embedding problem, the only input data one has available are edge connectivities. Although the high dimensional vectors $\{\mathbf{y}_i\}_{i=1}^N$ are generally non-existent, it is possible to cast the optimization problem (1) entirely in terms of the inner products $\mathbf{y}_i^\top\mathbf{y}_j$ for all $i, j \in \{1, \ldots, N\}$. This brings to bear many of the merits of kernel methods entailing computations involving only inner products of transformed feature vectors, $\phi(\mathbf{y})$, namely $k_{ij}(\mathbf{y}_i, \mathbf{y}_j) = \phi^\top(\mathbf{y}_i)\phi(\mathbf{y}_j)$. The problem is now reduced to selection of an appropriate kernel matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$ with $[\mathbf{K}]_{ij} := k_{ij}(\mathbf{y}_i, \mathbf{y}_j)$. A few examples of K include:

i. The doubly-centered dissimilarity matrix $\mathbf{K} = -(1/2)\mathbf{J}\boldsymbol{\Delta}^{(2)}\mathbf{J}$ where $[\boldsymbol{\Delta}^{(2)}]_{ij}$ denotes the squared geodesic distance between nodes i and j, or any other dissimilarity metric on the graph, and $\mathbf{J} := \mathbf{I} - N^{-1}\mathbf{1}\mathbf{1}^\top$ denotes the centering operator [4]. In this case, $\mathbf{K}$ is reminiscent of the kernel adopted by classical MDS.

ii. The Penrose-Moore pseudoinverse of the graph Laplacian i.e., $\mathbf{K} = \mathbf{L}^\dagger$, where $\mathbf{L} := \mathbf{A} - \mathbf{D}$, $\mathbf{A} \in \{0, 1\}^{N\times N}$ denotes the binary graph adjacency matrix, and $\mathbf{D} := \text{diag}(\mathbf{A}\mathbf{1})$. It turns out that $\mathbf{L}^\dagger$ admits an intuitive interpretation as a similarity matrix based on random walk distances on graphs [9].

iii. Matrix $\mathbf{K} = \mathbf{A}\mathbf{A}^\top$, where $\mathbf{A} \in \{0, 1\}^{N\times N}$, and $[\mathbf{A}\mathbf{A}^\top]_{ij}$ counts the number of single-hop neighbors shared by nodes i and j.

Note that determination of $\{\mathcal{N}_i\}_{i=1}^N$ in LLE is rather costly and entails $\mathcal{O}\left(qN^2\right)$ complexity with $q \gg$. The proposed graph embedding strategy overcomes this bottleneck by setting $\mathcal{N}_i$ to the single-hop neighborhood of node i in $\mathcal{G}$. Let $\mathbf{Y}_i := [\phi(\mathbf{y}_1^i), \ldots, \phi(\mathbf{y}_{d_i}^i)]$ collect the "virtual" transformed vectors associated with the single-hop neighbors of i, where d_i denotes its degree. Furthermore, letting $\mathbf{w}_i := [w_{i1}, \ldots, w_{id_i}]^T$, the LS fit (1) can be be written as

$$\mathbf{w}_i = \underset{\{\mathbf{w}:\ \mathbf{1}^\top\mathbf{w}=1\}}{\arg\min} \mathbf{w}^\top\mathbf{K}_i\mathbf{w} - 2\mathbf{w}^\top\mathbf{k}_i \tag{3}$$

where $\mathbf{K}_i := \mathbf{Y}_i^\top\mathbf{Y}_i$ and $\mathbf{k}_i := \mathbf{Y}_i^\top\phi(\mathbf{y}_i)$ are submatrices of $\mathbf{K}$ indexed by the elements of $\mathcal{N}_i$. Using the theory of Lagrange multipliers, one can readily solve for $\mathbf{w}_i$ in (3). For large-scale graph embedding, the weight computation step can be easily parallelized over clusters of computing nodes. Each subproblem entails $\mathcal{O}(d_i^3)$ complexity, which is manageable because typically $d_i \ll N$.

The low-dimensional graph embedding can be obtained from the reconstruction weights via (2), by solving for

$$\underset{\left\{\begin{smallmatrix}\mathbf{X} \\ \sum_{i=1}^N \mathbf{x}_i=\mathbf{0} \\ \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i\mathbf{x}_i^\top=\mathbf{I}\end{smallmatrix}\right\}}{\arg\min} \text{Tr}\left[\mathbf{X}^\top(\mathbf{I}-\mathbf{W})^\top(\mathbf{I}-\mathbf{W})\mathbf{X}\right] \tag{4}$$

where $\mathbf{W}^\top := [\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_N]$, $\tilde{w}_{ij} = w_{ij}$ if $j \in \mathcal{N}_i$, otherwise $\tilde{w}_{ij} = 0$, $\mathbf{X}^\top := [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, and Tr(.) denotes matrix trace. Note that the $N \times p$ embedding matrix $\mathbf{X}$ consists of the 2nd to the $(p+1)$th least dominant eigenvectors of $(\mathbf{I}-\mathbf{W})^\top(\mathbf{I}-\mathbf{W})$.

Remark 1. Typically $|\mathcal{N}_i| \ll N$, hence $\{\tilde{\mathbf{w}}_i\}_{i=1}^N$ are sparse. One can augment (3) with a sparsity-promoting penalty term so that neighborhood selection is data-driven; that is, $\tilde{\mathbf{w}}_i = \underset{\tilde{\mathbf{w}}}{\arg\min}\ \tilde{\mathbf{w}}^\top\mathbf{K}_i\tilde{\mathbf{w}} - 2\tilde{\mathbf{w}}^\top\mathbf{k}_i + \gamma\|\tilde{\mathbf{w}}\|_1 \quad \text{s.t.} \quad \mathbf{1}^\top\tilde{\mathbf{w}} = 1$, where $\gamma > 0$.

Remark 2. For large graphs, $\mathbf{X}$ can be efficiently computed via the generalized power method (a.k.a. orthogonal iterations [18]) entailing $\mathcal{O}(pN^2)$ complexity. Furthermore, orthogonal iterations are amenable to decentralized implementation; see e.g., [16].

Although this approach preserves the local graph topology within single-hop neighborhoods, the spectral decomposition is not scalable to larger networks. Moreover for large network visualization tasks, one is often more interested in graph embeddings that easily convey global properties such as node hierarchy or community structure. Towards addressing these issues, the next section advocates a novel framework for embedding graphs under hierarchical constraints.

4. CENTRALITY CONSTRAINTS

In settings involving web-scale networks, meaningful graph embedding for visualization purposes calls for emphasis of key structural information over traditional aesthetic requirements. Several centrality measures on networks have been proposed for quantifying the

relative importance of nodes over their peers e.g., node degree, closeness centrality, and PageRank [19]. Their capacity to succinctly establish hierarchical rankings of nodes (and edges) in a graph makes them attractive for graph embedding tasks. Let $\mathcal{C}(\mathcal{G}) := \{c_i\}_{i=1}^N$ denote the set of centralities of $\mathcal{G}$, with c_i representing the centrality measure of node i. The goal here is to determine the embedding $\{\mathbf{x}_i\}_{i=1}^N$ that effectively "preserves" the centrality ordering in $\mathcal{C}(\mathcal{G})$.

The advocated approach involves incorporating centrality constraints in the final step (4); that is,

$$\begin{aligned} \underset{\mathbf{x}_1,\ldots,\mathbf{x}_N}{\arg\min} \quad & \sum_{i=1}^N \left\| \mathbf{x}_i - \sum_{j=1}^N w_{ij}\mathbf{x}_j \right\|_2^2 \\ \text{s. to} \quad & \|\mathbf{x}_i\|_2^2 = f^2(c_i),\ i = 1,\ldots,N. \end{aligned} \tag{5}$$

where $f(c_i)$ is a monotone decreasing function of c_i to ensure that more central nodes are placed closer to the center. The dropped $\mathbf{0}$-mean constraint can be compensated for by a post-processing centering operation upon determination of $\{\hat{\mathbf{x}}_i\}_{i=1}^N$.

Note that (5) is non-convex with no global optimality guarantees. However, it decouples over vectors $\{\mathbf{x}_i\}_{i=1}^N$ which motivates a block coordinate descent (BCD) approach [3]. The optimization variables are divided into N blocks with $\mathbf{x}_i$ corresponding to block i. As a result, each iteration r cycles through all blocks solving

$$\begin{aligned} \mathbf{x}_i^r = \underset{\mathbf{x}}{\arg\min} \quad & \left\| \mathbf{x} - \sum_{j<r} w_{ij}\mathbf{x}_j^r - \sum_{j>r} w_{ij}\mathbf{x}_j^{r-1} \right\|_2^2 \\ \text{s. to} \quad & \|\mathbf{x}\|_2^2 = f^2(c_i). \end{aligned} \tag{6}$$

per block i. 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 $\lambda \geq 0$ denoting a Lagrange multiplier, then

$$\mathbf{x}_i^r = \underset{\mathbf{x}}{\arg\min} \quad \|\mathbf{x} - \mathbf{v}_i^r\|_2^2 + \lambda(\|\mathbf{x}\|_2^2 - f^2(c_i)) \tag{7}$$

whose solution is

$$\mathbf{x}_i^r = \frac{\mathbf{v}_i^r}{1+\lambda}. \tag{8}$$

Upon substituting (8) into the equality constraint in (6), one obtains the closed-form per-iteration update

$$\mathbf{x}_i^r = \begin{cases} \frac{\mathbf{v}_i^r}{\|\mathbf{v}_i^r\|_2} f(c_i), & \text{if } \|\mathbf{v}_i^r\|_2 > 0 \\ \mathbf{x}_i^{r-1}, & \text{otherwise.} \end{cases} \tag{9}$$

Letting $\mathbf{X}^r$ denote the embedding matrix after r BCD iterations, the operation $\mathbf{X} = (\mathbf{I} - N^{-1}\mathbf{1}\mathbf{1}^\top)\mathbf{X}^r$ centers $\{\mathbf{x}_i^r\}_{i=1}^N$ to the origin in order to satisfy the shift invariance property of the embedding. Algorithm 1 summarizes the steps outlined for the centrality-constrained graph embedding scheme. 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, and the kernel matrix $\mathbf{K}$.

Algorithm 1 Kernel-Based Graph Embedding

Input: $\mathcal{G}, \mathcal{C}(\mathcal{G}), \mathbf{K}, \epsilon, p$
for $i = 1 \ldots N$ (in parallel) **do**
  Set $\mathcal{N}_i$ to single-hop neighbors of i
  Extract $\mathbf{K}_i$ and $\mathbf{k}_i$ from $\mathbf{K}$
  Solve $\mathbf{w}_i = \arg\min_{\mathbf{w}} \ \mathbf{w}^\top\mathbf{K}_i\mathbf{w} - 2\mathbf{w}^\top\mathbf{k}_i$ s. t. $\mathbf{1}^\top\mathbf{w} = 1$
  Set $w_{ij} = 0$ for $j \notin \mathcal{N}_i$
end for
Initialize $\mathbf{X}^0$, $r = 0$
repeat
  $r = r + 1$
  for $i = 1 \ldots N$ **do**
    Compute $\mathbf{x}_i^r$ according to (9)
    $\mathbf{X}^r(i,:) = (\mathbf{x}_i^r)^T$
  end for
until $\|\mathbf{X}^r - \mathbf{X}^{r-1}\|_F \leq \epsilon$
$\mathbf{X} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top)\mathbf{X}^r$

5. NUMERICAL TESTS

This section presents numerical tests conducted on randomly generated networks to demonstrate the effectiveness of the developed kernel-based graph embedding approach. The Watts-Strogatz model was used to generate synthetic random graphs [26]. Given a prescribed number of nodes N, the average degree $\bar{d}$, and a parameter $\beta \in [0,1]$, the Watts-Strogatz model constructs a $\bar{d}$-regular ring lattice, and rewires each edge with probability β. Twenty Watts-Strogatz graphs were generated this way with

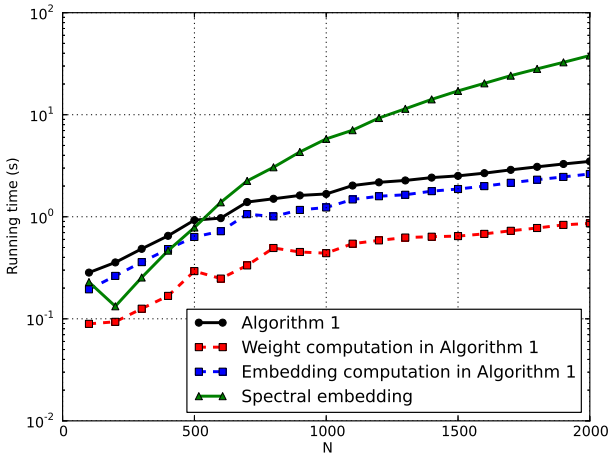

Fig. 1. Comparison of running times of Algorithm 1 on Watts-Strogatz graphs of increasing size benchmarked against spectral embedding.

$N \in \{100n\}_{n=1}^{20}$, $\bar{d} = 4$, and $\beta = 0.3$. Hierarchical structure was captured through closeness centrality, which is commonly defined as $c_i := 1/\left(\sum_{j\in\mathcal{V}} d_{ij}\right)$, where d_{ij} denotes the geodesic distance between nodes i and j. The centralities were accordingly transformed as follows

$$f(c_i) = \left(\frac{c_{\max} - c_i}{c_{\max} - c_{\min}}\right) \tag{10}$$

where $c_{\max} := \max_i \ c_i$, and $c_{\min} := \min_i \ c_i \ \ \forall i = 1,\ldots,N$.

First, the running time of Algorithm 1 was assessed based on an implementation done on a commodity personal computer. Figure 1 plots running times for the weight computation step, the embedding step, and the total running time (all in seconds) against the number of nodes in the graph. In order to facilitate comparisons with contemporary structure-preserving methods, the spectral embedding algorithm of [23] was run for each graph. The plot depicts the corresponding running times alongside those obtained by running Algorithm 1. It turns out that Algorithm 1 outperforms spectral embedding by up

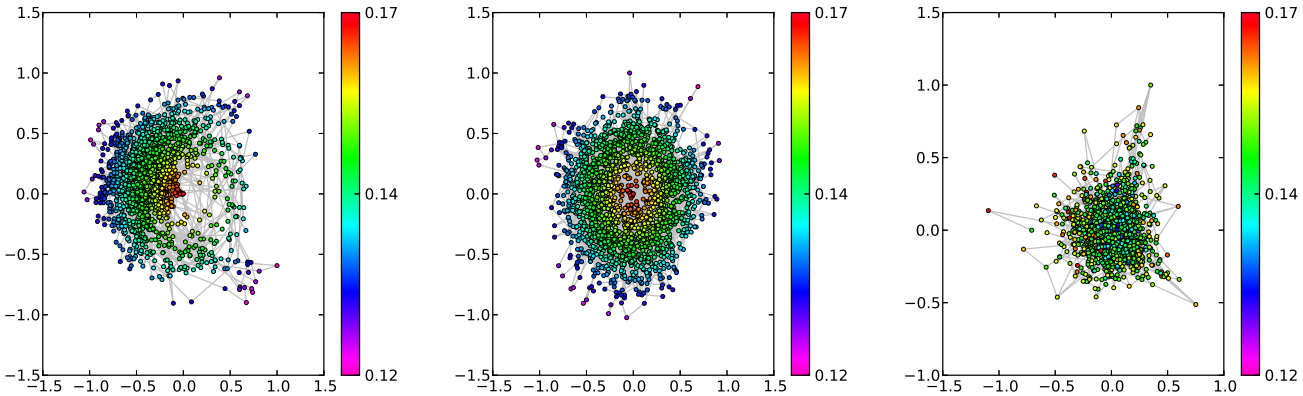

Fig. 2. Graph embeddings for a Watts-Strogatz graph with $N = 2,000$: (a) Centrality-constrained embedding with $\mathbf{K}_1 = -1/2\mathbf{J}\mathbf{\Delta}^{(2)}\mathbf{J}$; (b) Centrality-constrained embedding with $\mathbf{K}_2 = \mathbf{A}\mathbf{A}^\top$; and (c) Centrality-agnostic embedding based on kernel matrix $\mathbf{K}_1$. The color bar maps node colors to varying centrality values.

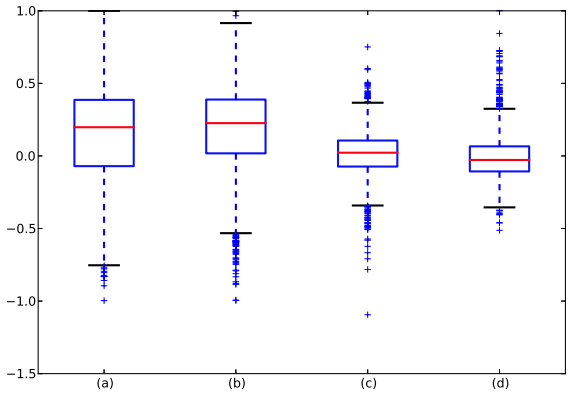

Fig. 3. Boxplots (a) and (b) depict the spread of the centrality-constrained embedding, while (c) and (d) are based on the centrality-agnostic approach. It is clear that the higher appeal of the centrality-based visualizations is due to the wider point spread, and fewer "outliers".

to an order of magnitude when $N > 500$ nodes, hence it is well-motivated for large network visualization tasks.

Figure 2 depicts visualizations of the Watts-Strogatz network obtained by setting $N = 2,000$. In Figure 2 (a) and (b), centrality-constrained embeddings are plotted with kernel matrices $\mathbf{K}_1 = (-1/2)\mathbf{J}\mathbf{\Delta}^{(2)}\mathbf{J}$ and $\mathbf{K}_2 = \mathbf{A}\mathbf{A}^\top$, respectively. The appeal for centrality-constrained embeddings is clear when compared with Figure 2(c) which depicts a "centrality-agnostic" graph embedding using $\mathbf{K}_1$. Here, the final weight preservation step entailed spectral decomposition of $(\mathbf{I} - \mathbf{W})^\top(\mathbf{I} - \mathbf{W})$, consistent with the original LLE algorithm. It is clear that even for a moderately sized synthetic graph, little meaningful information can be conveyed visually from the centrality-agnostic embedding. For instance, it is not obvious how an analyst would discern which nodes are most accessible to peers, or those whose removal would compromise the rate of information propagation over the network.

Assessment of the quality of a network visualization is generally a non-trivial and subjective task. In this paper, a rather simple scheme based on the spread of the embedding per coordinate is advocated. Figure 3 depicts boxplots of the embeddings in Figures 2 (a) and (c). Boxplots (a) and (b) depict the distribution of the centrality-constrained embedding coordinates, whereas (c) and (d) depict the centrality-agnostic case. The plot shows that the centrality-based embedding yields a wider point spread and fewer outliers, which translates into higher visual appeal.

6. CONCLUSION

This paper develops a novel graph embedding scheme based on LLE, a popular nonlinear manifold learning approach. LLE thrives in settings where data are sampled from low-dimensional manifolds, where the goal is to preserve the local neighborhood structure. Consistent with contemporary graph embedding approaches, one typically only has access to the graph topology. It is shown that a well-defined graph kernel suffices to compute the reconstruction weights per node, circumventing the need to build feature vectors. It turns out that the cost incurred by determination of the neighborhoods is markedly reduced by selection of single-hop neighbors per node. Furthermore, the weight determination step can be easily parallelized with closed-form solutions per subproblem.

The need to capture meaningful structural patterns e.g., node hierarchy, motivates a centrality-constrained approach which entails constraining the weight preservation step. The resulting problem is separable and the advocated approach resorts to BCD iterations with closed-form per-block updates. The combination of parallel weight determination and low cost BCD iterations leads to a more appealing embedding algorithm than contemporary approaches based on PCA and MDS for large network visualization. Tests conducted on several small-world networks demonstrate the appeal of the developed centrality-aware algorithm for capturing hierarchical patterns in network visualizations. Due to space limitations, tests on big web-scale networks are deferred to a more comprehensive version of the paper.

7. REFERENCES

[1] 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.

[2] B. Baingana, and G. B. Giannakis, "Centrality-constrained graph embedding," *Proc. of the 38th Intern. Conf. on Acoust., Speech and Signal Proc.*, Vancouver, Canada, May 26–31, 2013.

[3] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1999.

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

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

[6] 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.

[7] 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, Jan. 2012.

[8] 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.

[9] 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, Mar. 2007.

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

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

[12] 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, Apr. 2002.

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

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

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

[16] S. Jain, S.-J. Kim, and G. B. Giannakis, "Backhaul-constrained multi-cell cooperation leveraging sparsity and spectral clustering," *arXiv preprint arXiv:1409.8359*, Sep. 2014.

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

[18] D. Kempe and S. McSherry, "A decentralized algorithm for spectral analysis," *Journal of Computer and Systems Sciences*, vol. 74, pp. 70–83, Feb. 2008.

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

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

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

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

[23] M. Saerens, F. Fouss, L. Yen, and P. Dupont, "The principal components analysis of a graph, and its relationships to spectral clustering," *Proc. of the 15th European Conference on Machine Learning*, Pisa, Italy, Sep. 20–24, 2004.

[24] 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.

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

[26] D. J. Watts and S. H. Strogatz, "Collective dynamics of small world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[27] 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, Sep. 2007.

[28] 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.