Exploring the Distribution of Local Neighborhood Structures in Large Networks

Shenghui Cheng, Claudia Dahl, Joachim Giesen, Philipp Lucas, Klaus Mueller

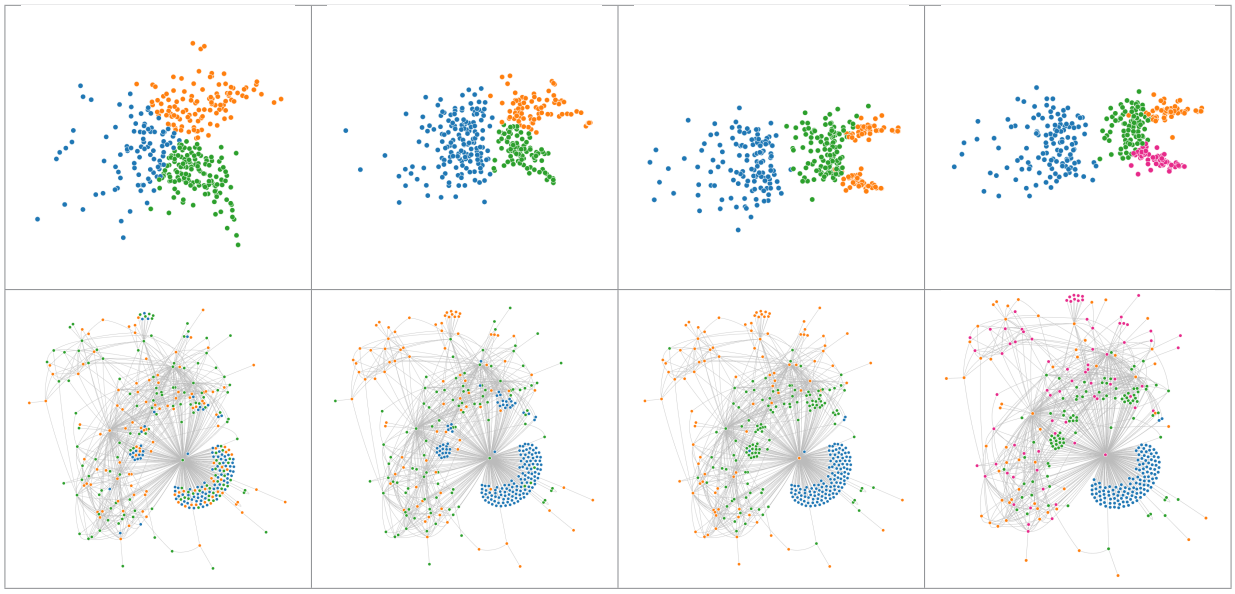

Fig. 1. Each column shows a MDS-plot of feature vectors that describe the local neighborhood structure of the nodes in the TWITTER network (top) and a force-directed network layout of the same network (bottom). Colors correspond to a k-means clustering for $k = 3$ or 4 (right) on the local feature vectors.

Abstract—Network exploration techniques aim at identifying features like clusters, central nodes or motifs in networks. Motifs are local sub-networks of a network that appear more frequently than expected. We introduce a new type of local neighborhood structure that is spectrally defined, i.e., based on the eigenvalues and eigenvectors of some matrix associated with the network, and show that well established visual analytics techniques for exploring high-dimensional point clouds provide an effective means for the exploration of the distribution of these neighborhood structures. Experiments on real world social networks demonstrate that our approach is indeed capable of revealing interesting neighborhood structures that are not easily accessible otherwise.

Index Terms—Large networks, local neighborhood structure, spectral embedding, spectral clustering, high dimensional data.

1 INTRODUCTION

Networks arise almost everywhere, e.g. as social networks, road networks, protein-protein interaction networks or as hyperlink networks of documents. Exploratory network analysis aims at identifying characteristic features of a network $G = (V, E)$, where V is a finite set of nodes and E is a set of links that connect the nodes. Well studied network features include

1. clusters, i.e., a partitioning of the node set V into groups of similar nodes,
2. central nodes, and
3. motifs, i.e., small sub-networks of G that are more frequent than expected when compared to some given random graph model.

The type of feature that we study in this paper is similar in spirit to motifs, namely we are looking for typical neighborhood structures in a network. On the technical side our approach is close to very successful approaches for node clustering and for computing node centrality indexes that both can be tackled with spectral techniques. These techniques build on the eigenvalues and eigenvectors of some suited matrix associated with a given network.

Eigenvector centrality uses the eigenvector to the largest eigenvalue of the adjacency matrix to define a centrality score for each node, i.e., the centrality score of node x_i is defined as

$$c_i = \frac{1}{\lambda} \sum_{j \in N(i)} v_j,$$

where λ is the largest eigenvalue of the adjacency matrix of the network G and $v = (v_1, \ldots, v_n)$ is the corresponding eigenvector. Here we assume that the nodes of G are indexed from 1 to $|V|$, and $N(i)$ is the set of indexes of the neighbors of the vertex x_i in G. Slight modifications of the Eigenvector centrality are Katz centrality and the

- *Claudia Dahl, Joachim Giesen and Philipp Lucas are with Friedrich-Schiller-Universität Jena, Germany.*
- *Shenghui Cheng and Klaus Mueller are with SUNY Stony Brook, USA and SUNY, Korea.*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014; date of publication xx xxx 2014; date of current version xx xxx 2014.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

PageRank. See [2] for a comprehensive overview on graph centrality measures.

Spectral clustering, see for example [8], is a popular family of graph clustering techniques. In a nutshell it works as follows: take some symmetric matrix associated with the network, popular choices are graph Laplacians, and compute the top-k eigenvectors $v_1,\ldots,v_k$ of this matrix, i.e., the eigenvectors to the k leading eigenvalues $\lambda_1 \geq \ldots \geq \lambda_k$. The eigenvectors are then used to embed the nodes of G in the Euclidean space $\mathbb{R}^k$ by the following mapping

$$x_i \mapsto (v_{1i},\ldots,v_{ki}) \in \mathbb{R}^k,$$

i.e., the node x_i is mapped to the i'th components of the top-k eigenvectors. The embedded nodes can then be clustered using some Euclidean clustering algorithm. The standard here is the k-means clustering algorithm. Here we are going to use a variant of the spectral embedding from above that is called diffusion maps [12]. For diffusion maps the Euclidean distance in the embedding space can be related to a diffusion distance in the original network. We will provide a brief review of this technique later in the paper.

The new idea that we want to present here is using Euclidean embeddings as they are used in spectral clustering for the identification of typical neighborhood structures in a network. For that we just consider the k-nearest neighbors[1] of any node in the graph in the embedding into Euclidean space and determine the local shape of this neighborhood by a local principal component analysis (PCA). The eigenvalues and eigenvectors that are determined by the local PCA characterize the local geometry, e.g., if all eigenvalues are of the same magnitude, then the neighborhood looks spherical, while it looks ellipsoidal otherwise. The local eigenvectors determine a local coordinate system. The local geometry that is encoded in the eigenvalues and eigenvectors of the local PCA can be represented by $2k$-dimensional feature vector. The entirety of the feature vectors for all nodes of the network provides yet another point cloud.

Our main contribution is the introduction of visual analytics techniques for exploring these feature vector point clouds and thus the local neighborhood structure of a network. Experiments with real world social networks show that our approach allows to detect novel features in these networks.

2 The processing pipeline

Here we describe in some detail the processing pipeline that computes a local neighborhood feature vector for every node in the network. Starting point of our pipeline is the adjacency matrix A of an undirected network $G=(V,E)$. If the number of nodes in the network is n, then A is a symmetric $n\times n$ matrix whose entry a_{ij} is 1 if $\{i,j\}\in E$, i.e., the nodes x_i and x_j are connected, and 0 otherwise.

2.1 Similarity matrix

The first step in the pipeline is computing a similarity matrix S from the adjacency matrix A. Our requirements on a similarity matrix are that it is a symmetric, positive semidefinite $n\times n$ matrix, whose entry s_{ij} is a measure for the similarity of nodes x_i and x_j. Popular similarity matrices are

1. $S=A^2$, i.e., the matrix product of the adjacency matrix with itself. In this case two nodes are considered similar, if they share many common neighbors. Note that two nodes can be similar even if they are not connected.

2. $S=A\cdot A^2$, i.e., the point-wise product of the first similarity matrix with the adjacency matrix. Here two nodes are considered similar if they are connected and share many common neighbors.

3. $S=(s_{ij})$ with $s_{ij}=\exp(-\lambda c_{ij}^2)$, where c_{ij} is the graph distance between the nodes x_i and x_j, i.e., the length of a shortest path connecting x_i and x_j in the network, and $\lambda>0$ is some scaling parameter.

[1] Please excuse the heavy overloading of the character k that of course takes different values when computing the top-k eigenvalues, the k-means clustering, and the k-nearest neighbors.

2.2 Spectral embedding

The second step in the pipeline transforms the network into a Euclidean point cloud via a spectral embedding. Nadler et al. [12] have suggested diffusion maps as an embedding technique, where the Euclidean distance in the embedding space has an interpretation as a diffusion distance in a network whose weighted adjacency matrix is given by the similarity matrix S. For the definition of a diffusion map Nadler et al. consider the matrix $D^{-1}S$, where $D=(d_{ij})$ is the diagonal matrix with entries $d_{ii}=\sum_{j=1}^n s_{ij}$. Note that the matrix $D^{-1}S$ is stochastic, i.e., it row sums are 1, and thus induces a Markov chain. Note though that $D^{-1}S$ is not necessarily symmetric anymore. The eigenvalues and the left- and right eigenvectors of $D^{-1}S$ can be obtained from the eigenvalues $\lambda_0\geq\lambda_1\geq\ldots\geq\lambda_{n-1}$ and the corresponding eigenvectors $v_0,\ldots,v_{n-1}$ of the adjoint matrix $D^{-1/2}SD^{-1/2}$. The left eigenvectors of $D^{-1}S$ are given as $\phi_i^T=v_i^TD^{1/2}$ and the right eigenvectors are given as $\psi_i=v_iD^{-1/2}$, respectively, for the eigenvalues λ_i. That is, $\phi_i^TD^{-1}S=v_i^TD^{-1/2}S=v_i^TD^{-1/2}SD^{-1/2}D^{1/2}=\lambda_iv_i^TD^{1/2}=\lambda_i\phi_i^T$ and similarly for the right eigenvectors. For $k\leq n$ the rank k diffusion map (embedding) at time $t\in\mathbb{N}$ applied to vertex $j\in V$ is defined as

$$\psi_t^{(k)}(x)=\big(\lambda_1^t\psi_1(x),\ldots,\lambda_k^t\psi_k(x)\big),$$

where $\psi_i(x)$ is the x-component of the vector ψ_i. The diffusion distance between two nodes in the network is defined as

$$\begin{aligned} d^2(x_i,x_j) &= \big\|p(t,x|x_i)-p(t,x|x_j)\big\|_{\phi_0}^2 \\ &= \sum_{x\in V}\big(p(t,x|x_i)-p(t,x|x_j)\big)^2\phi_0(x)^{-1}, \end{aligned}$$

where $p(t,x|x_i)=e_i^T(D^{-1}S)^t$, i.e., multiplying the i'th standard basis vector from the left to the t'th power of the stochastic matrix $D^{-1}S$, which corresponds to t steps in the Markov chain with transition matrix $D^{-1}S$, where at $t=0$ the whole probability mass is concentrated in the vertex x_i. The diffusion distance thus compares the probability mass distribution after t time steps when the initial distribution is concentrated at x_i and x_j, respectively.

Nadler et al. prove that

1. $d^2(x_i,x_j)=\big\|\psi_t^{(n-1)}(x_i)-\psi_t^{(n-1)}(x_j)\big\|$
2. $\Big|d^2(x_i,x_j)-\big\|\psi_t^{(k)}(x_i)-\psi_t^{(k)}(x_j)\big\|^2\Big|\leq\lambda_{k+1}^2\left(\frac{1}{\phi_0(x_i)}-\frac{1}{\phi_0(x_j)}\right)$

That is, the Euclidean distance of the mapped vertices recovers their diffusion distance for rank $n-1$ embeddings, and it is a good approximation for rank k embeddings if the eigenvalues decay quickly.

2.3 Local neighborhood features

In the third step of the pipeline we compute local neighborhood features for the nodes of the network. Once the vertices of the network have been embedded by the mapping

$$x_i\mapsto\psi_t^{(k)}(x_i)\in\mathbb{R}^k$$

we can compute their ℓ nearest neighbors in Euclidean distance. Let x be a node and $N(x)$ be the set of its ℓ nearest neighbors. We compute a local PCA, i.e., local for the node x, for the vectors

$$\psi_t^{(k)}(y)-\psi_t^{(k)}(x),\quad y\in N(x)$$

which gives us ℓ eigenvalues $\mu_1\geq\mu_2\geq\ldots\geq\mu_\ell$ and their corresponding eigenvectors $u_1,\ldots,u_\ell$. The feature vector that we assign to x is now simply given as

$$\big(\mu_1,\ldots,\mu_\ell,\omega_1,\ldots,\omega_\ell\big),$$

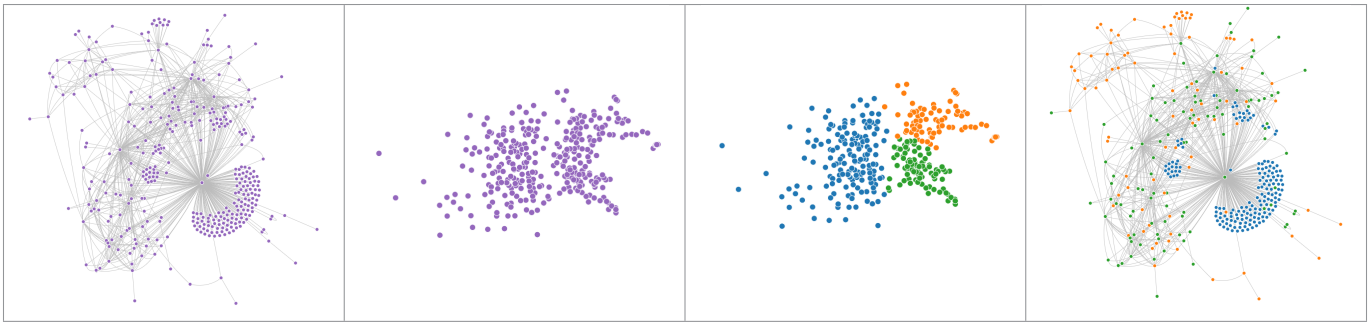

Fig. 2. Left: Force-directed graph layout for the TWITTER data set. Middle/left: MDS plot of the local feature vectors for the same network. Middle/right: The same MDS plot as before where the nodes are colored according to a k-means clustering. Right: The same graph layout as on the left where the nodes are colored as before.

where $\omega_j = e_j^T u_j$, i.e., the cosine of the angle between u_j and the j'th standard basis vector $e_j \in \mathbb{R}^\ell$ that has the entry 1 at index j and 0 otherwise. The feature vector describes the local geometry in the neighborhood of the embedding of x. The μ's describe the shape of the neighborhood, i.e., if all μ's are of the same magnitude, then the neighborhood is spherelike, while it looks ellipsoidal otherwise. The ω's just describe the rotation of the local coordinate system given by the vectors u with respect to some fixed reference coordinate system. Here the reference coordinate system is given by the standard basis vectors of $\mathbb{R}^\ell$.

3 VISUAL EXPLORATION OF THE DISTRIBUTION OF LOCAL NEIGHBORHOOD STRUCTURES

In Section 2 we showed how a feature vector that characterizes the local neighborhood can be computed for every node of a given network. The challenge now is to visualize the network and the gamut of feature vectors despite their high dimension in a way that a user can identify typical local neighborhood structures. We apply well established techniques to tackle that problem, namely

1. we use a force-directed layout to present the network to the user,
2. use multidimensional scaling (MDS) to reduce the effective dimension of the feature vectors to two while maintaining a good approximation of the pairwise distances,
3. provide an exploratory, interactive user interface. It shows side-by-side a force-directed node-link layout of the network on the left and a MDS plot of the feature vectors on the right. Built-in brushing and linking of these plots improves the usability. The user can also change parameters of the processing pipeline on-the-fly and immediately see the changed visualization, and
4. apply k-means clustering on the local feature vectors to extract structural types of local neighborhoods.

3.1 Drawing the graph

Many layout methods exist for graph visualization [5]. We aim at obtaining an overview of the network at hand and choose a force-directed layout to show the nodes and their connecting links. In order to avoid cluttering if appropriate we do not draw the links as straight line segments but as Bézier curves as in [13], see Figure 2(left) for an example.

3.2 Visualizing the local feature vectors

Numerous methods have been proposed for high dimensional data visualization, e.g., scatter plot matrices [4], parallel coordinates [6], generalized barycentric coordinates [3], and others. Multidimensional scaling (MDS) [1] is a good method that allows to quickly gauge the original similarity and structure of a high dimensional data set. Thus, we choose it to visualize the local feature vectors.

3.3 Cluster analysis

As the MDS plot shows a very condensed view on the inherently high dimensional local feature vectors, it may be hard to distinguish clusters visually on that level. Therefore, we first cluster the original feature vectors and then color the nodes of the graph and the points in the MDS plot accordingly. Here we chose k-means clustering [9]. Note, that this is different from the classic spectral clustering approach [8] where a clustering algorithm is applied directly to the spectral embedding of the nodes of the network. See Figure 2(right) for an example.

3.4 Interactivity

Just identifying clusters in the MDS plot of the local feature vector is not enough. The user needs to link it to the original network. Hence, we implemented brushing and linking [7] such that when the user selects a certain area in the right-hand plot, corresponding nodes in the original graph on the left are highlighted by drawing them larger. See Figure 4.

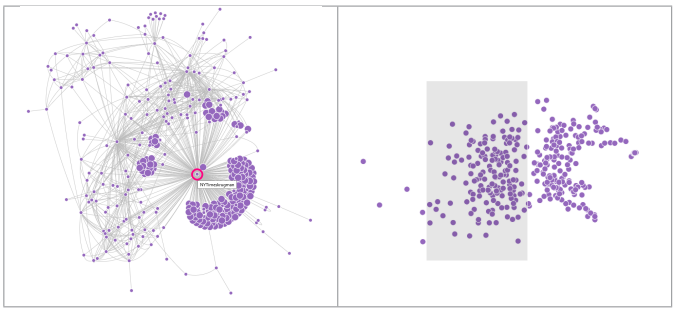

Fig. 4. The link interaction between the original network and embedding points. Points selected on the right are highlighted on the left. To support semantic analysis the user can also read the label of a node by hovering over it with the mouse, see the marked node NYTIMESKRUGMAN on the left.

No single set of parameters used in the pipeline of Section 2 will suffice for all data sets. Also, the dimension of the spectral embedding should be at most the number of nearest neighbors used for the local PCA, because otherwise the vectors to the nearest neighbors do not span the whole embedding space which renders a meaningful comparison of the orientation of the local coordinate systems impossible. Hence, we face a trade-off decision: on the one hand the spectral embedding gets more accurate with higher dimension, i.e., it preserves the diffusion distance better, but on the other hand, we must choose a reasonably small number of nearest neighbors for the local PCA, as that number directly effects the scale of the neighborhood structures we are able to identify. Therefore, we take an exploratory approach to find a suitable set of parameters for a given network and let the

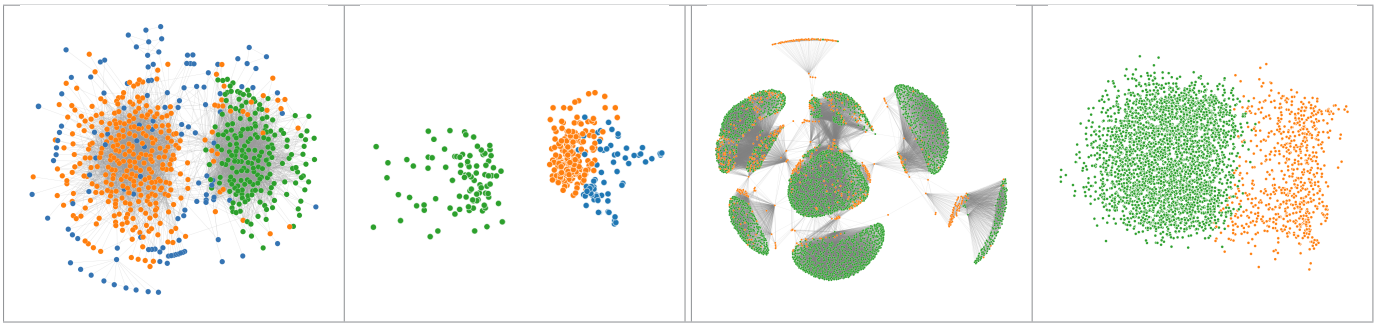

Fig. 3. From left to right: Graph layout for the TWITTER(EXT) data set, MDS plot of the local feature vectors for the same data set, graph layout for the FACEBOOK data set, and MDS plot of the local feature vectors for the same data set.

user change them as required. Specifically, the user can dynamically change

- the dimension of the embedding of the graph,
- the number of neighbors to be used for the local PCA, and
- the number of clusters to be found in the k-means clustering (see the following section).

All changes have an immediate effect on the visualization.

4 CASE STUDIES

In the sections below we demonstrate using the example of two real world social networks (TWITTER and FACEBOOK) that our method is capable of revealing typical local neighborhood structures. For both case studies we used the similarity matrix $S = A^2$ (see Section 2).

4.1 Twitter

The TWITTER network data set contains 350 nodes and 835 links, where nodes represent twitter users and a link is present if a user "follows" another user, see [10] for more details. We also considered an extended data set TWITTER(EXT) that contains 640 nodes and 7988 links, where nodes represent twitter users who used the hashtag "My2K" and a link is present if a user "follows", "replies-to" or "mentions" another user, see again [10] for more details.

For the results that we report here we chose a six dimensional spectral embedding and used 15 nearest neighbors for the local PCA. Only for Figure 1 that shows the effect of varying the parameters of our pipeline we chose 4 (in three dimensions), 10 (in six dimensions) and 25 (in 15 dimensions), respectively, neighbors for the local PCA. There we observe that as we increase the size of the local neighborhood, the MDS-projection spreads and splits into different clusters more clearly.

For the TWITTER data set the feature vectors clearly separate into clusters, see Figure 1. One cluster contains nodes (shown in light blue) that are almost only connected to the central node NYTIMESKRUGMAN. Another cluster contains nodes in the boundary of the network (shown in yellow), and a third cluster contains nodes in between (shown in green). The last cluster can be split into two, but a direct interpretation in the graph layout seems difficult. A k-means clustering on the full dimensional feature vectors confirms our visual finding. Moreover, thanks to brushing and linking, as well as consistent coloring across the MDS-plot and the graph layout, we can now relate the neighborhood structure to the original network.

We get similar results for the data set TWITTER(EXT), see Figure 3(left). Here the feature vectors clearly separate into two clusters that interestingly correspond to two clusters in the original network. That means these two clusters can also be described by local neighborhood structures which cannot be seen at all just from the graph layout.

4.2 Facebook

The FACEBOOK network data set [11] contains 4039 nodes and 88234 links. The results that we report for this data set were obtained from an eight-dimensional spectral embedding and a choice of eight nearest neighbors for the local PCA.

The MDS plot, see Figure 3(right), shows that the feature vectors clearly separate into two pronounced clusters, each corresponding to a certain local neighborhood structure. From the graph layout we see that nodes of the first cluster correspond to nodes that belong to clusters in the original network, while the nodes in the second cluster are in-between clusters. We call nodes of the latter type as *interface nodes*. See again Figure 3(right).

5 CONCLUSION

In this paper we have proposed a visual analytics method for finding typical neighborhood structures in large networks. Preliminary results show that our method indeed allows to detect local structures that would be hard to find otherwise. In future work we want to explore different similarity matrices, e.g., similarity based on the shortest path distance.

REFERENCES

[1] I. Borg and P. Groenen. *Modern multidimensional scaling theory and applications*. New York: Springer, second edition, 2005.

[2] S. P. Borgatti and M. G. Everett. A graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, 2006.

[3] S. Cheng and K. Mueller. Improving the fidelity of contextual data layouts using a generalized barycentric coordinates framework. In *Proceedings of IEEE Pacificvis*, pages 295–302, April 2015.

[4] J. A. Hartigan. Printer graphics for clustering. *Journal of Statistical Computation and Simulation*, 4(3):187–213, 1975.

[5] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.

[6] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of IEEE Visualization*, pages 361–378, 1990.

[7] P. Isenberg and D. Fisher. Collaborative brushing and linking for co-located visual analytics of document collection. In *IEEE-VGTC Symposium on Visualization 2009*, 2009.

[8] U. v. Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

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

[10] B. S. Marc A. Smith, Lee Rainie and I. Himelboim. Mapping twitter topic networks: From polarized crowds to community clusters, 2014.

[11] J. Mcauley and J. Leskovec. Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):4, 2014.

[12] B. Nadler, S. Lafon, R. Coifman, and I. G. Kevrekidis. Diffusion maps-a probabilistic interpretation for spectral embedding and clustering algorithms. In *Principal manifolds for data visualization and dimension reduction*, pages 238–260. Springer, 2008.

[13] D. W. Ulrik Brandes. Using graph layout to visualize train interconnection data. *Journal of Graph Algorithm and Application*, 2000.