Identifying the Skeptics and the Undecided through Visual Cluster Analysis of Local Network Geometry

Shenghui Cheng[a,d,e], Joachim Giesen[c], Tianyi Huang[a,d], Philipp Lucas[c], Klaus Mueller[b]

[a] *Westlake University, Hangzhou, China*
[b] *Stony Brook University, New York, US*
[c] *Friedrich-Schiller-Universität Jena, Jena, Germany*
[d] *Westlake Institute for Advanced Study, Hangzhou, China*
[e] *Research Center for the Industries of the Future, Hangzhou, China*

Abstract

By skeptics and undecided we refer to nodes in clustered social networks that cannot be assigned easily to any of the clusters. Such nodes are typically found either at the interface between clusters (the undecided) or at their boundaries (the skeptics). Identifying these nodes is relevant in marketing applications like voter targeting, because the persons represented by such nodes are often more likely to be affected in marketing campaigns than nodes deeply within clusters. So far this identification task is not as well studied as other network analysis tasks like clustering, identifying central nodes, and detecting motifs. We approach this task by deriving novel geometric features from the network structure that naturally lend themselves to an interactive visual approach for identifying interface and boundary nodes.

Keywords: Graph/Network Data, High Dimensional Data Visualization, Visualization in Social and Information Sciences, Data Clustering Coordinated and Multiple Views

Email address: chengshenghui@westlake.edu.cn (Shenghui Cheng)

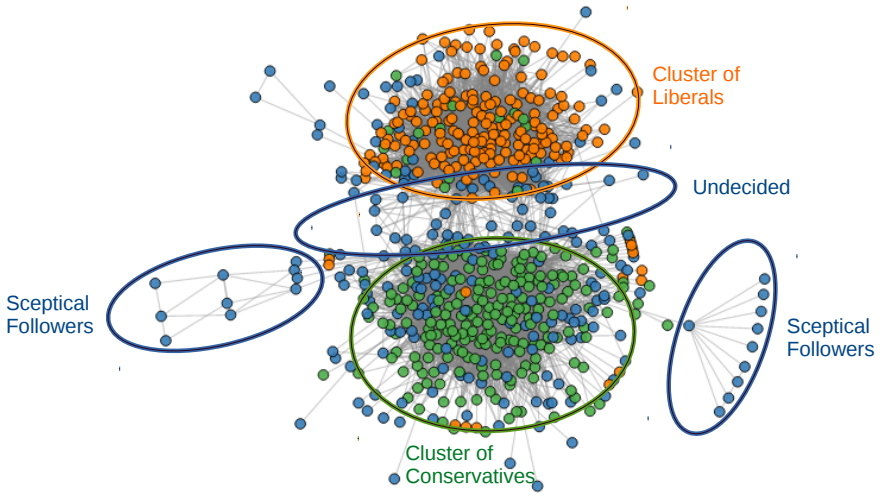


Figure 1: The TWITTER POLARIZED CROWD network is a typical example of a discussion network on a political topic as it features two groups of users that form distinct discussion groups, a liberal and a conservative group, that hardly interact with each other. For applications like voter targeting the most interesting nodes in such a network are the nodes at the interface between the two groups or nodes that are only loosely linked to one of the groups, because it is more likely that they can be influenced by a political campaign.

1. Introduction

Customer targeting in marketing is an important problem is our social life. Voter targeting is a representative instance of the customer targeting. It has become an important tool at latest in the 2004 presidential election and is now used heavily by both the Democrats as well as the Republicans. Marketing activities can be roughly divided into two groups, namely (1) identifying target markets and market segments by means of market analysis, and (2) applying methods for influencing customer behavior by providing product information and/or product promotions. Of course marketing activities incur costs and thus have to planned carefully. For instance, assuming that providing product information to customers incurs costs, then it makes sense to address only those customers or groups of customers that have a high enough chance to be convinced of buying the advertised product or service. Hence, identifying such customers or groups has become an important marketing activity. In our time and

age, marketing techniques are so advanced that targeting individual customers instead of customer groups or segments has become technically and financially feasible. Besides classical socio-demographic information social networks have become an interesting data source that can be used for targeting customers.

Classifying the customers into undecided and skeptics makes sense in the broader marketing context. It is useful, for instance, in saturated markets with high brand loyalty like the tobacco market. Smokers with high brand loyalty are hard to reach by marketing campaigns. More likely to be reached by such campaigns are undecided smokers who switch between two or three brands, and skeptics who either smoke infrequently or mostly smoke cigarettes from the same brand, but frequently also from other brands.

Here we address the problem of identifying both marketing targets, the 'undecided' and the 'skeptics', from clustered social network data. A clustered network exhibits several densely connected groups with significantly fewer edges across the groups. Since both classes, the 'undecided' and the 'skeptics', are not clear cut, the problem is a prime candidate for a visual analytics approach that allows to visually identify the questionable nodes in the network and check if they are really members of the target classes. Throughout the rest of the paper we want to adopt the more technical terms of interface nodes for the 'undecided' and boundary nodes or outliers for the 'skeptics'. We describe and discuss a set of geometric features that can be efficiently computed from similarity information among nodes in a network. The features naturally lend themselves to an interactive visual approach for identifying interface and boundary nodes. We also describe how to derive similarity information from the incidence information in social networks and validate our approach and the accompanying visual exploration tool on the Twitter Polarized Crowd network and the Facebook Ego Networks data sets. Figure 2 shows the different visual explorations used in our work and their corresponding effects.

Organization of the paper. This paper is organized as follows: In the next section we position our paper in the context of related work. In Sections 3.1 and 3.2

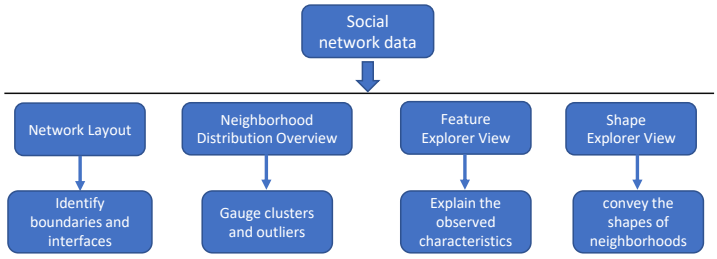


Figure 2: The different visual explorations and their corresponding effects.

we describe the ideas underlying our new geometric features and show how they can be computed. We have implemented an interactive tool that supports the feature based identification of interface and boundary nodes from structural social network data. The tool and the rationale behind the implemented views and interactions are described in Section 4. We apply our approach and tool in two case studies that are described in Section 5, and conclude the paper with Section 6 that summarizes our results.

2. Related work

Pretorius, Purchase and Stasko [27] give a comprehensive overview of tasks for multivariate network analysis. Abstractly speaking, a task is an analytic activity on the combination of an *entity* and a *property* of that entity. Network entities are according to Lee et al. [18] nodes, edges (or links), paths of nodes and edges, and whole networks. Network properties fall into two classes *structural* or *topological* properties and *attributes* associated with nodes and edges. Hence also network analysis tasks can be distinguished as either *structure-based*, or *attribute-based*. Here we want to focus only on structural properties and structure-based tasks. Connectivity tasks are a subset of structure-based tasks that according to Pretorius et al. include among others clustering-tasks and bridge-tasks that aim at finding *bridges* and *articulation points* in networks. Here we also want to add boundary/outlier-tasks for finding nodes that are

only loosely connected to clusters. Clustering, bridge- and boundary/outlier-tasks are all *analytical identify* tasks in the sense of Valiati et al. [1].

In this paper we address variants of clustering and certain bridge- and outlier-tasks: Finding bridges or articulation points (aka cut vertices) is a classical topic in network analysis and graph algorithms [12]. In our toy example of Section 4 (see Figure 4), Node 4 is an articulation point since his removal increases the number of connected components of the network (from one to two). In larger, node clustered networks there are typically more nodes than just a single one that have a significant number of connections in more than one cluster. These nodes are at the interface of the clusters, but are no longer articulation points, see for example Figure 1. Still, in terms of our intended application of identifying undecided consumers from network data we also want to detect these nodes. Another difference to the classical work is that we address the problem not only on primary information, i.e., just the adjacency matrix of the network, but also on derived, secondary information, namely node similarities. Similarly for boundary nodes and outliers. Based on primary information, boundary nodes and outliers are nodes with a small degree, i.e., a small number of incident edges. Using also secondary information allows to detect also nodes that do not necessarily have small degree, but are dissimilar to almost all the other nodes. In a marketing context boundary nodes and outliers correspond mostly to skeptics.

A natural question to ask at this point is, why is an interactive visual approach necessary for identifying interface and boundary nodes, especially since we are going to introduce automatically computable features that are meant to identify these nodes. The simple answer is, like the notion of a cluster itself also the notions of interface nodes and outliers are fuzzy. Hence fully automatic methods are not well suited for identifying them reliably. The real benefit of these features is providing filters to be interactively used by the analyst for shrinking the search space of all nodes to likely candidates of outliers and interface nodes. The candidates can then be inspected further by the analyst.

For completeness, we also give here a brief overview of some tools and tech-

niques that have been designed for supporting structure based, multivariate networks analysis tasks, although none of them supports our specific goal of identifying interface and boundary nodes.

Nishikawa and Matter [25] list 28 efficiently computable network properties most of which are of a spectral nature. They use these properties for embedding the nodes of a given network into 28-dimensional space and search for group structures (clusters) in the resulting point cloud by standard clustering techniques as well as by an interactive, visual inspection of randomly chosen two-dimensional projections. Cheng et al. [11][10] apply visualization to show the connection activities in torus network but is not for general network.

Wong et al. [33] introduce another structural network property dubbed as graph signatures. A signature of degree d is defined for each node as the vector $(n_1, \ldots, n_d)$, where n_i is the number of the nodes at distance i from the node. Graph signatures can be computed efficiently by using breadth-first-search. The signature vectors can be embedded into the plane by multi-dimensional scaling. Among some other tasks, graph features aid in finding articulation points, but are in general not geared towards identifying outliers and interface nodes.

Wattenberg [32] introduces the PivotGraph tool for analyzing multivariate networks. PivotGraph uses a simple grid-based approach to focus on the relationship between node attributes and connections. Its interaction approach is derived from an analogy with methods seen in spreadsheet pivot tables and in online analytical processing (OLAP).

The multidimensional nature of multivariate network data suggests the use of standard multidimensional visualization techniques like scatterplot matrices (SPLOMs) and parallel coordinate plots (PCPs) for their analysis. The GraphDice tool [3] is adapting a SPLOM for the analysis of multivariate network data. In an overview plot matrix one node-link plot for each pairwise combination of attributes is shown, i.e., a scatter plot matrix of the attributes together with plotted edges. The user can select one plot as the main plot which is then enlarged. The GraphDice tool extends the ScatterDice tool [14] for navigating and exploring multidimensional tables and thus inherits most of

its rich interaction capabilities.

Viau et al. [31] go even further in adapting standard multidimensional visualization techniques for analyzing multivariate network data. They introduce the parallel scatterplot matrix (P-SPLOM) as a unification of scatter plot matrices (SPLOMs) and parallel coordinate plots (PCPs), together with smooth transitions between them. Additionally, they propose to use hybrid network layouts, i.e., a mix of attribute-driven layouts with force-directed and manual layouts of the nodes, in order to provide more freedom and customizability to the exploring user. Radviz [6] [9] and data context map [7] are also important techniques to visualize multidimensional data with the help of colorization [8] or other enhancements [35], but both of them are not suitable for graph data.

Vehlow et al. [30] propose a neat method of visualizing overlapping communities and the fuzzyness of community assignment at different levels of details. Similiar in purpose Wu et al. [34] describe an approach to interactive visual summary of communities in large networks. They visually encode each community as a polygon. Boundary nodes that are not clearly assigned to any community are drawn individually in between the polygons. However, both, Wu and Vehlow, require a priori community information for each node, e.g. from some community detection algorithm. In our work we rely on no such information but introduce novel geometric node features based on the local neighborhood of nodes. We then use a visual analytics approach to explore these features in order to identify interface and boundary nodes.

3. Preliminary

Our work is based on the existing local neighborhood features and similarity matrices from networks. Therefore, we review them in this section.

3.1. Local neighborhood features

Here we motivate and describe the construction of geometric features for identifying interface and boundary nodes in social networks that greatly improve

on ad hoc features like node degrees. Keep in mind, though, that these features are not complete in the sense that they enable a fully automatic identification of these nodes. They are intended to be used interactively as filters to reduce the search space of all nodes in the network.

3.1.1. Input data

We consider some similarity matrix as our basic data structure. The similarity matrix is typically derived from heterogeneous data sources like socio-demographic data, online activity data, social network data and many more. This paper is not concerned with computing similarity scores from primary data sources except for Section 3.2, where we derive a similarity matrix from social network data. Deriving similarity measures is a standard machine learning task. A common approach is turning data points into feature vectors and then defining similarity as the dot product between feature vectors. The similarity matrices that have been computed that way, i.e., every similarity score is a dot product, are also called Gram matrices. Often Gram matrices are not computed explicitly from feature vectors, but implicitly from a kernel function on the data [28]. In the latter case, the Gram matrix is also called a kernel matrix. Gram matrices are not only symmetric, but also positive semi-definite. For technical reasons that will become apparent later, here we also want to assume that our similarity matrices are symmetric and positive semi-definite.

3.1.2. Deriving structure from similarity

Machine learning can be used for deriving structure from data encoded in a similarity matrix. A popular structure that can be computed from similarity matrices are clusters, i.e., a partitioning the data into groups such that the inner-group similarities are large while the intra-group similarities are low. In a marketing context clustering is mostly referred to as market segmentation.

Here we are interested in a secondary structure: A cluster is a fuzzy concept. While the assignment of some of the data points to a cluster is obvious for a human observer it might be quite dubious for others. Our goal is identifying

data points whose cluster assignments are not so obvious. In the marketing context these are the customers whose assignment to a market segment might be loose and that are thus prime candidates to be targeted in market campaigns that aim at promoting a different segment, for instance smokers whose brand loyalty is not very high or undecided voters in election campaigns.

Many machine learning techniques, for instance linear support vector machines [28] or k-means clustering, do not work on abstract similarity matrices but on Euclidean point clouds. To make these techniques amenable to similarity matrices the latter are often transformed into Euclidean point clouds such that the Euclidean distance after the transformation approximates the (dis-)similarity well. This is also the approach that we want to pursue here, namely deriving geometric features that support the identification of interface and boundary nodes in a clustered social network from an Euclidean embedding of the nodes.

3.1.3. Spectral embedding

Spectral embedding is a popular technique for embedding similarity matrices into Euclidean space such that the Euclidean distance of the points associated with similar data points is small, and large for dissimilar nodes. Compared with other mapping techniques, like MDS, spectral embedding can better present the local manifold of the samples and thus better mine the clusters in data. In our spectral embedding, the data are encoded now in a Euclidean point cloud instead of a similarity matrix. So far we have considered the encoding

$$data \;\rightarrow\; similarity \quad matrix,$$

now we consider the encoding

$$data \;\rightarrow\; similarity \quad matrix \;\rightarrow\; Euclidean \quad point \quad cloud.$$

Given a sample matrix $X = [x_1; x_2; ...; x_n]$, the spectral embedding firstly constructs a neighbor-based similarity graph associated with a weighted adja-

cency matrix W. Each element $w_{i,j}$ in W is defined as follows.

$$w_{i,j} = \begin{cases} e^{-\frac{d_{i,j}^2}{\sigma^2}} & x_i \in \mathcal{N}_j \quad or \quad x_j \in \mathcal{N}_i, \\ 0 & otherwise, \end{cases}$$

where $\mathcal{N}_i$ is the set of the nearest neighbors of x_i, $d_{i,j}$ is the distance between x_i and x_j, and σ is a free parameter. Then the features which well represent the manifold of this graph can be extracted by the following problem:

$$\mu = \arg \min_{\substack{\mathbf{1}D\mu=0 \\ \mu^T\mu=1}} \frac{\frac{1}{2}\sum_{i,j}(\mu_i - \mu_j)^2 w_{i,j}}{\sum_i D_{ii}\mu_i^2},$$

where D is a diagonal matrix with the ith diagonal element as $d_{i,i} = \sum_j w_{i,j}$. With the normalization of $\mu^T\mu = 1$, the optimal solution μ to this problem is the normalized eigenvector corresponding to the smallest nonzero eigenvalue of $D^{-1}L$, where $L = D - W$ is a Laplacian matrix [29]. Although, this eigenvector can well represent a part of manifold structures in the above graph, we can not well visualize the samples by only this eigenvector. As the suboptimal solutions which correspond to a few small eigenvalues, some of the next normalized eigenvectors also contain useful partitioning information and thus can be used to our visualization. After the spectral embedding, the point for each x_i is defined as $p_i = (\mu_{i1}, \ldots, \mu_{id})$. The computation complexity for spectral embedding is $O(n^3)$.

The point $p_i \in \mathbb{R}^d$ can now be clustered by the k-means algorithm or any other geometric clustering algorithm, or be classified by a linear support vector machine, provided we also have class labels for the points.

3.1.4. Using geometry beyond clustering

The key idea that we want to explore here is that the geometric information encoded in a Euclidean point cloud should be useful for more than linear classification or clustering. We are especially interested in the local distribution of the points within the point cloud. Our working assumption is that the points

at the boundaries or in between clusters are locally distributed differently than points deep within clusters.

A first, simple measure for the local distribution of the points is the average distance of a point to its k nearest neighbors. It turns out that this measure does indeed provide useful information about interface and boundary nodes, but is far from a reliable, complete feature for identifying these nodes.

As a slightly more complex measure for the local distribution of the points, that works much better than the simple average distance, we propose to fit an ellipsoid to the k nearest neighbors of each of the points such that the vectors from the points to their neighbors span the whole space $\mathbb{R}^d$. Hence, it is necessary tho choose $k \geq d$. The shape of the ellipsoids depends on the distribution of the k-nearest neighbors and thus can serve as an approximation of the shape of the neighborhood.

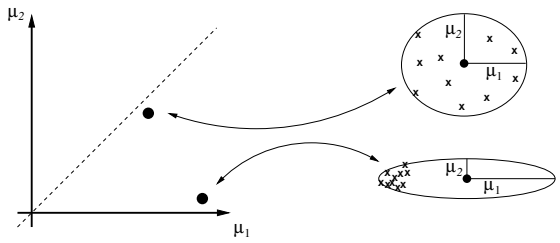


Figure 3: The shape of a local neighborhood is approximated by the ellipsoid E_i that is associated with the covariance matrix $Q_i Q_i^\top$. The shape of the ellipsoid is determined by the reciprocal eigenvalues of $Q_i Q_i^\top$. Here are shown two examples: (1) a spherelike neighborhood, where $\mu_1 \approx \mu_2$, and (2) an elongated neighborhood, where $\mu_1 \gg \mu_2$.

Fitting ellipsoids can be computed by a local principal component analysis (PCA). Let

$$\{p_{i_1}, \ldots, p_{i_k}\}$$

be the set of the k-nearest neighbors of p_i, and define

$$q_{ij} \;=\; p_{i_j} - p_i, \qquad \text{for } \; j = 1, \ldots, k.$$

The q_{ij} are organized in a $k \times n$ matrix Q_i, whose columns are the vectors q_{ij}.

Here n is the number of nodes. The covariance matrix $Q_i Q_i^\top$ is symmetric and positive definite with rank k, and thus

$$E_i \;=\; \left\{x \in \mathbb{R}^k \;:\; x^\top Q_i Q_i^\top x = 1\right\}$$

is an ellipsoid. The semi-principal axes of E_i are given by the eigenvectors of $Q_i Q_i^\top$ and their lengths are given by reciprocals of the corresponding eigenvalues. Hence, if $0 < \lambda_{i1} \leq \ldots \leq \lambda_{ik}$ are the eigenvalues of $Q_i Q_i^\top$, then

$$0 \;<\; \mu_{ik} := 1/\sqrt{\lambda_{ik}} \;\leq \ldots \leq\; \mu_{i1} := 1/\sqrt{\lambda_{i1}}$$

are the lengths of the semi-principal axes of E_i. If all μ_{ij}'s are of the same magnitude, then the ellipsoid is spherelike, while its prolated or oblated otherwise, see Figure 3.

3.1.5. Local neighborhood features

We want to use the local information that is encoded in the ellipsoids E_i that are associated with the points p_i for distinguishing points deeply within clusters from points on their boundaries or in between clusters. The ellipsoids are up to rotations completely determined by the eigenvalues λ_{ij}, $j = 1, \ldots, k$ or the square roots of their reciprocals μ_{ij}. Hence, we assign the feature vector

$$\left(\mu_{i1}, \ldots, \mu_{ik}\right) \in \mathbb{R}^k$$

to the point p_i and thus to the i-th data point.

Of course, we can also cluster the feature vectors using for instance again k-means clustering, but most likely with a different value for k than for clustering the points p_i. Since the feature vectors are associated with the points that themselves are associated with the data points, we have a second clustering of the data points. In a marketing context, the first clustering is based on the Euclidean point representation of the customers and corresponds to a market segmentation, while the second clustering is based on the feature vectors

and indicates the degree to which a customer belongs to its assigned market segment. It is important to remember that the feature vectors can always be complemented by other features, most notably the assignment of the items to a cluster.

3.2. Similarity matrices from networks

So far we have started our discussion with a similarity matrix as our basic data structure. Here we show how to derive such a similarity matrix just from the structure of a network. For that purpose, let $G = (V, E)$ be a network, whose vertex set is denoted as V and whose edge set is denoted as E. In a social network an edge encodes some social interaction between the two incident nodes.

Let A be the adjacency matrix of the network, i.e., an $n \times n$-matrix, if there are $|V| = n$ nodes, whose entry a_{ij} is 1 if $\{i, j\} \in E$, and 0 otherwise. Adjacency matrices are by construction symmetric and could in principle serve as similarity matrices, but in this case similarity would be just a binary feature that tells if two nodes socially interact, or not. Furthermore, A is not necessarily positive semi-definite, though this is mostly a technical detail. A much better choice for a similarity matrix is A^2, i.e., the matrix product of the adjacency matrix with itself. The interpretation here is that two nodes are more similar to each other, if they have more neighbors in common, i.e., a larger number of nodes they both interact with. Also, A^2 is symmetric and positive semi-definite. Hence, it is not surprising that A^2 is a well established similarity measure in social network analysis, where it is known as *structural equivalence* [20]. Of course, there are also other possibilities to define symmetric, positive semi-definite similarity matrices from adjacency matrices, e.g., based on shortest paths, but A^2 is an intuitive and convenient choice especially in the context of our geometric working assumption that points at the boundaries or in between clusters are distributed differently than points deep within clusters. Here the points correspond to the nodes in the network that have been spectrally embedded into Euclidean space. Let us check the working hypothesis on the following toy example.

Toy example. Our toy example is a network with ten nodes: two disjoint cliques with four nodes each, one node connected to all nodes in both cliques, and one node connected to two nodes in one clique. See Figure 4

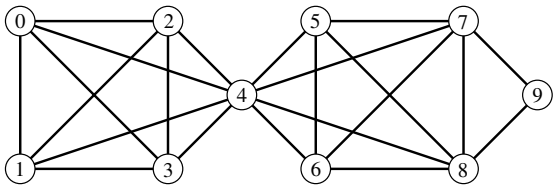


Figure 4: This toy example features two cliques/clusters (Nodes 0,1,2,3, and 5,6,7,8, respectively), one node on the interface between the two clusters (Node 4), and one node that is loosely coupled to the second cluster (Node 9). The aim of our work is identifying the interface and loosely coupled nodes.

The adjacency matrix A of our toy network is given as

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

and thus the corresponding similarity matrix A^2 is given as

$$A^2 = \begin{pmatrix} 4 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 0 \\ 3 & 4 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 0 \\ 3 & 3 & 4 & 3 & 3 & 1 & 1 & 1 & 1 & 0 \\ 3 & 3 & 3 & 4 & 3 & 1 & 1 & 1 & 1 & 0 \\ 3 & 3 & 3 & 3 & 8 & 3 & 3 & 3 & 3 & 2 \\ 1 & 1 & 1 & 1 & 3 & 4 & 3 & 3 & 3 & 2 \\ 1 & 1 & 1 & 1 & 3 & 3 & 4 & 3 & 3 & 2 \\ 1 & 1 & 1 & 1 & 3 & 3 & 3 & 5 & 4 & 1 \\ 1 & 1 & 1 & 1 & 3 & 3 & 3 & 4 & 5 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 1 & 1 & 2 \end{pmatrix}$$

We are simply using the top two eigenvectors of A^2 for embedding the nodes of the network into the plane. The embedding after adding some jitter is shown in Figure 5

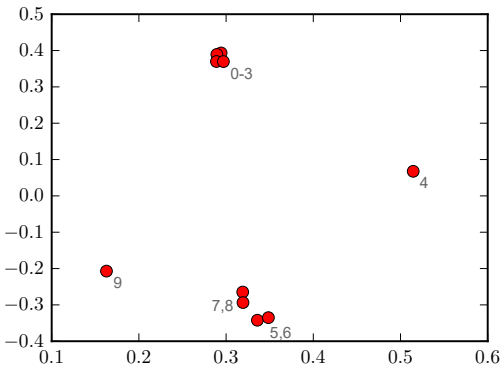


Figure 5: Using the top two eigenvectors of A^2 for embedding the nodes of the toy network.

As can be seen in Figure 5, the interface Node 4 and the boundary Node 9 can be clearly detected by the average distance to their nearest neighbors. Furthermore, the fairly large degree of Node 4 renders it unlikely that this node is a boundary node. Hence, we can easily classify Node 4 as an interface node

and Node 9 as a boundary node. In more complex networks, like the TWITTER POLARIZED CROWD network, see Figure 1, the classification is not that easy anymore and having the more complex geometric features at our disposal becomes beneficial.

4. Visual analytics framework

We have implemented an interactive tool that allows the exploration of primary features (clusters, market segments) and secondary features (neighborhood shape) in network data. The tool supports four fully linked views that we describe in the following.

4.1. Views

The four views of our tool (Network Layout, Neighborhood Distribution Overview, Feature Explorer, and Shape Explorer) are summarized in Figure 6. We should point out here that the contribution of this paper is not providing novel visualizations, but demonstrating that well established, standard visualization techniques in conjunction with our geometric features can be an effective means for reaching our goal of identifying interface and boundary nodes in clustered social networks.

Network Layout View. Many different graph layout strategies are known that allow to represent a network as a node-link diagram in the plane, where nodes are typically represented as disks and links as straight or curved line segments, see for example [2] for an overview of different layout strategies and goals. The Network Layout view of our tool features a stress minimization layout that aims at preserving the node similarity, i.e., tries to place similar nodes close to each other. Additionally, we have experimented with a force directed layout [2] that aims at avoiding visual clutter and edge crossings, and a backbone layout [26] that has been designed for separating different communities in social networks. It turned out that the latter two layouts do not support the identification of boundary and interface nodes well, see Figure 14 (a). Always keep in mind that

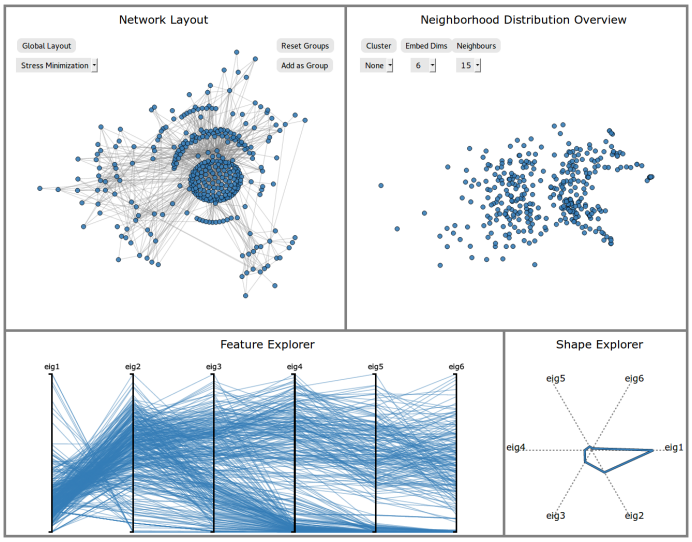


Figure 6: The four views implemented in our tool for identifying interface and boundary nodes: Network Layout, Neighborhood Distribution Overview, Feature Explorer, and Shape Explorer

this is the only view that does not show the feature vectors, but the original network.

Neighborhood Distribution Overview View. Neighborhood distribution overview is used to show the clusters and outliers in the high dimensional data. The entirety of local neighborhood feature vectors, see Section 3.1, that have been computed for the nodes of a given network can be considered a high dimensional point cloud. Numerous methods have been proposed for visualizing high dimensional point clouds, among them scatter plot matrices [15], parallel coordinates [16], and multidimensional scaling (MDS) [4] that are mentioned here as representatives for wholes classes of techniques that even can be combined. Since MDS allows to quickly gauge the structure, e.g., clusters and outliers, of a high dimensional data set we choose it as our primary view for visualizing the local neighborhood features.

Feature Explorer View. While MDS plots work well for detecting clusters and outliers they cannot explain which dimensions of the feature vectors are respon-

sible for the observed characteristics. However, feature explorer allows such insights by parallel coordinates that map high dimensional points to polylines. Hence, we chose to add it as an alternative view for visualizing the local feature vectors, i.e., the eigenvalues of the local covariance matrices. It should be noted that we scale the eigenvalues shown in this view such that the largest among the i-th eigenvalues in the data set is always set to 1 and the smallest is set to 0. Scaling the eigenvalues makes it easier to tell apart different clusters of the feature vectors and allows for an easier interaction, namely restricting the range of some of the eigenvalues for filtering, see Section 4.2. A drawback of scaling the eigenvalues is that the parallel coordinates view does not convey the shape of the local neighborhood properly.

Shape Explorer View. Neither the MDS plot nor the parallel coordinates plot allows to assess the shape of the local neighborhood directly. For that purpose, we add a shape explorer as a star plot view [17] to our tool that features a star plot of the local neighborhood features. Star plots immediately convey the shape of a local neighborhood, i.e., the fitted ellipsoid, fairly well. At one glance one can tell if the ellipsoid is more rounded or elongated. Remember that elongated ellipsoids hint at the interface or boundary nodes in the network, see also Figure 7.

The star plots in our tool are mostly used for groups of feature vectors and not for individual feature vectors, i.e., for selections or clusters, whose average is then shown in a star plot. For instance, the star plot shown in Figure 6 represents the average shape for all nodes in the network.

4.2. Interactions

Interactions are used to configure the visualizations and allow to explore the different neighborhood features for identifying interface and boundary nodes. The implemented interactions include the *selection* of nodes, *projections* onto a subset of the dimensions of the feature vectors, and *filtering* by these dimensions and other features like node degrees and the average distance to the k nearest

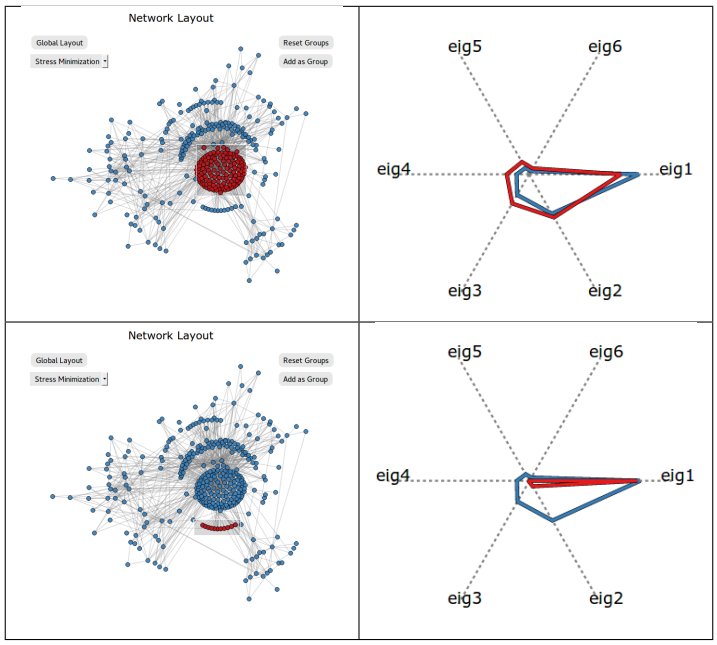


Figure 7: The selection interaction between the Network Layout view and the Shape Explorer view for two different selections in the Network Layout view. Apparently different groups of nodes in the Network Layout view have different local neighborhood shape characteristics as witnessed in the red star plots. Note that the average shape of the nodes selected on the left is much closer to the average shape (blue) than for the nodes selected on the right.

neighbors. Additionally, feature vectors can be clustered, and colored groups can be created from selections.

Selection. Nodes of the network can be selected using rectangular range queries either in the Network Layout view or in the Distribution Overview view. The selection also becomes active in the other views (brushing). For example in Figure 8 we show the link between a selection in the Distribution Overview view with all the other views, and in Figure 7 we demonstrate the link between a selection in the Network Layout view and the Shape Explorer view. To support semantic analysis a user can also read off the label of a node, if available, by hovering over it with the mouse.

Filtering. Filtering just means restricting the range of the nodes' attributes. In our case the attributes are node degree, average distance to the nearest

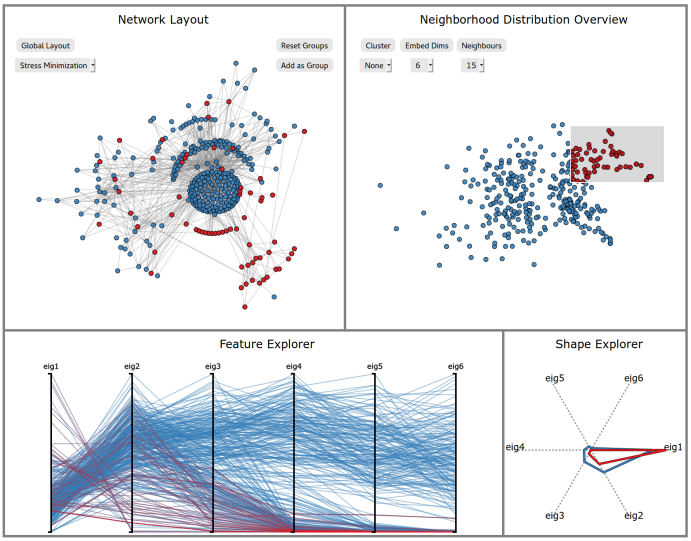


Figure 8: The selection interaction between the Distribution Overview view and all the other views. Points selected in the Distribution Overview view are highlighted in the Network Layout view and the Feature Explorer view. The Shape Explorer view features star plots for all nodes (blue) and the selected nodes (red).

neighbors, and the dimensions of the feature vectors, where every dimension is considered as an attribute of its own. Filtering the feature vector dimensions is enabled in our tool in the parallel coordinates plot of the Feature Explorer view, where one can select a range of values for the corresponding eigenvalues.

Projection. A projection is used for removing some of the dimensions from the local feature vectors. This interaction is possible in the Feature Explorer view and also affects the Distribution Overview view and the Shape Explorer view. It does not affect the Network Layout view since this view does not depend on the local feature vectors.

Clustering. For obtaining a first impression of a clustering structure within the distribution of the local feature vectors we provide the user with the choice of automatically clustering the feature vectors first. The nodes of the graph and the points in the MDS and parallel coordinates plots are then colored accordingly, see for example Figure 11 (b).

Our tool currently supports k-means clustering [19], where the value of k can be set interactively by the user. Typically, small values for k work well and one of the automatically computed clusters of the feature vectors corresponds already fairly well to the interface and boundary nodes. It is important to note, that the clustering of the feature vectors is different from the classic spectral clustering [21], where a geometric clustering algorithm is applied directly to a spectral embedding of the nodes of the network and not to secondary information like the feature vectors that we are using here.

5. Case Studies

We demonstrate our approach for finding interface and boundary nodes on the Twitter Polarized Crowd network and the Facebook Ego Networks data set.

For the spectral embedding of the networks into Euclidean space we used diffusion maps for the similarity matrices instead of directly using their eigenvectors. Diffusion maps have been suggested by Nadler et al. [24] who showed that the Euclidean distance of the embedded nodes has a nice interpretation as a diffusion distance in the network, if the embedding dimension equals the number of nodes in the network. They also prove that this diffusion distance is well approximated even for much smaller embedding dimensions.

The embedding dimension should be chosen in conjunction with the number k of nearest neighbors that are used for computing the neighborhood feature vectors. A default choice, that always gave good results also for other networks (for instance the Twitter Broadcast network that we used in all figures in the previous section), is choosing the embedding dimension as $k/2$. Note that the number of neighbors must always be larger than the embedding dimension.

The choice of the number of neighbors depends on the size of the network. Typically, this number should be smaller for smaller networks. In our case studies, any number between ten and twenty worked well in the sense that the results were not sensitive to the choice. Still, in our tool we provide the user

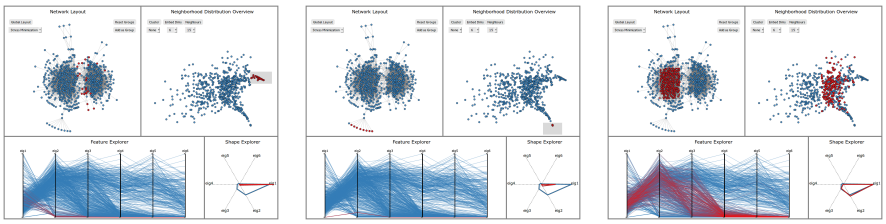


Figure 10: Three user types in the Twitter Polarized Crowd network. From left to right: undecided users (interface nodes), skeptics (boundary nodes), and stalwarts. Identifying these types is hard to come by using automatic techniques like clustering feature vectors, but requires the interactive features of our tool. The interactions here are straightforward selections either in the MDS or in the network layout view.

with the option to change the number of neighbors as well as the embedding dimension in a predefined range. For keeping the tool interactive, the results for the possible choices have to be precomputed.

5.1. Twitter Polarized Crowd

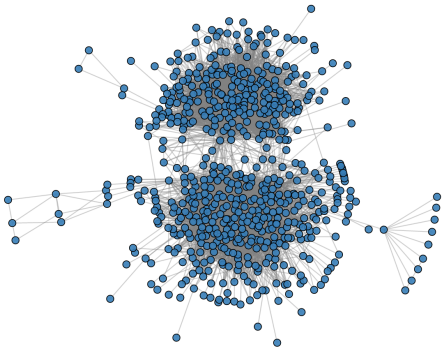


Figure 9: The Twitter Polarized Crowd network features two large and dense groups, liberals and conservatives, that have little connection to each other. They are mostly ignoring each other. Here shown is a stress minimization network layout.

The Twitter Polarized Crowd network has been introduced and discussed by Smith et al. [22] in their analysis of political conversations on Twitter. This network is a typical example of a discussion network on a divisive political topic as it features two groups of users that form distinct discussion groups, a liberal and a conservative group, that hardly interact with each other and

use different resources of information. Hence the tagging of this network as polarized crowds.

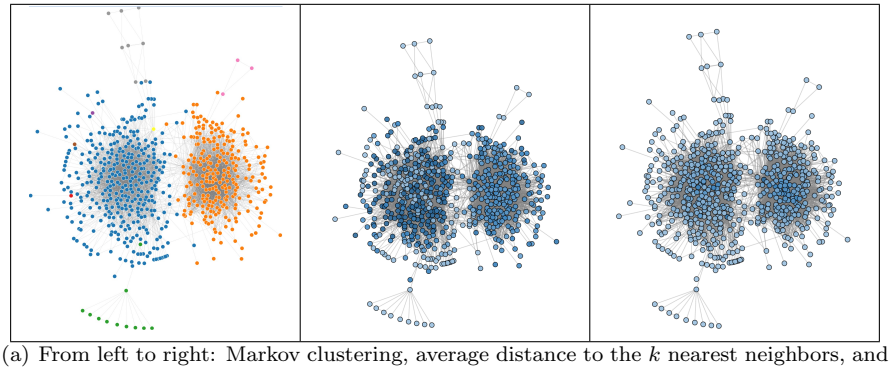
The network can also serve as an example for an increasing trend that people tend to surround themselves with other people who generally think alike. Thus, our similarity matrix A^2 (the squared adjacency matrix of the network) is a good measure for the polarization in a network, because nodes within clusters with respect to this similarity measure are hardly interacting with nodes outside the cluster by definition.

The TWITTER POLARIZED CROWD network has 640 nodes and $7,988$ links, where nodes represent twitter users who used the hashtag MY2K. This hashtag was promoted by the White House in 2012. It refers to an estimated $2,200 tax increase for middle class families if the Congress would not extend the Bush-era tax rates for families making $250,000 or less per year. A link is present in the network if a user *replies-to*, *mentions* or *follows* another user.

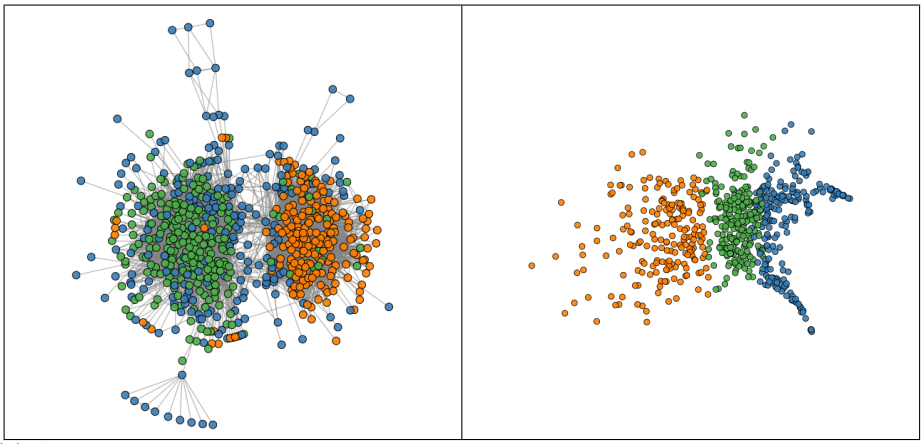
It can be seen in Figure 9 that the TWITTER POLARIZED CROWD network clearly separates into two clusters and some outliers. Clustering algorithms for graphs, for instance Markov clustering [13], easily detect these clusters, but are not capable of detecting interface and boundary nodes. Also, neither centrality scores, node degrees, nor the average distance to the k nearest neighbors in the spectral embedding of the network are particularly well suited features for identifying interface and boundary nodes, see Figures 11 (a) and 12 (left).

Identifying interface and boundary nodes. Applying a 3-means clustering to the neighborhood feature vectors, see Figure 11 (b), reveals a third cluster that roughly corresponds to interface and boundary nodes in a stress minimization layout of the network.

It should be pointed out here that it is remarkable that the first two clusters in the 3-means clustering correspond to the two groups, conservatives and liberals. The clustering has been computed on the neighborhood feature vectors and not on the spectral embedding of the network. That is, the two groups can also be distinguished by their neighborhood feature vectors which indicates a



(a) From left to right: Markov clustering, average distance to the k nearest neighbors, and node degree. Only the average distance aids in identifying interface and boundary nodes to some extent.



(b) An automatic 3-means clustering applied to the feature vectors shown directly in the network (left) and in a MDS plot of the feature vectors (right). Two clusters correspond to the two political groups, while a third cluster corresponds to interface and boundary nodes. The different clusters are labeled by the different colors.

Figure 11: The Clustering analysis on Twitter Polarized Crowd Network.

different communication behavior within each of the two groups. It is important to note that this difference in the neighborhood structure cannot be seen from the graph layout in Figure 9 (the same holds true also for other graph layouts), or from other means like centrality scores, see Figure 12 (left).

Automatically clustering the neighborhood feature vectors already gives a good first impression on the location of the interface and boundary nodes, but without distinguishing between them. An interactive exploration reveals more

detail. By selecting feature vectors in the Neighborhood Distribution Overview view or by selecting nodes in the Network Layout view one can easily distinguish three user types, see Figure 10:

Undecided (and nonpolitical) users: Members of this user type neither belong to the liberal nor to the conservative cluster. Their neighborhood structure is elongated since they have liberal as well as conservative users as their nearest neighbors. They are at the interface between the conservative and the liberal group. Typical node clustering algorithms are not able to detect this user type and put its members either into the liberal or into the conservative cluster, e.g. in Figure 10 (left) which shows a typical result of a node clustering algorithm.

Nonpolitical users: Members of this user type exhibit a similar neighborhood structure as the undecided users since their neighborhood structure is also elongated. The difference to the undecided users is that the nearest neighbors for members of this user type are either from the liberal or from the conservative cluster, but not from both, see Figure 10 (middle). The corresponding nodes are boundary nodes.

Stalwarts: Members of this user type have a spherical neighborhood structure, see Figure 10 (right). Their nearest neighbors belong to the same cluster, either liberal or conservative, as the members themselves. The members of this user type are typically also fairly central within their clusters, see Figure 11 (a) (right) for degree centrality scores. The corresponding nodes are neither interface nor boundary nodes.

A closer look. For providing ground truth we looked up some of the Twitter accounts that are present in the network (see Figure 12) and mostly have a high centrality score. It is important to note that the stress minimization layout only provides some indication, but not a ground truth for the classification of the nodes.

1. *bodiesoflight* is an esoteric Twitter account that essentially does not deal with political topics, and *hermanos* is a user who follows 429 other Twitter accounts that basically all deal with pop culture. Hence, it makes sense that

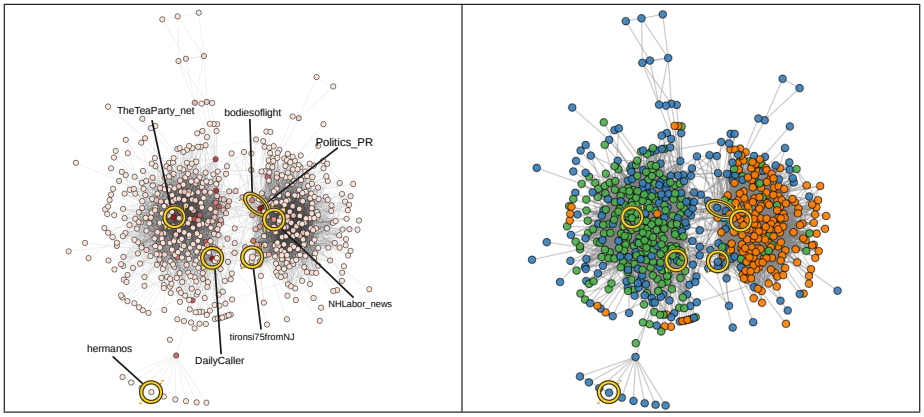


Figure 12: Highlighted are nodes deeply within the conservative and the liberal groups, nodes at the interface between the two groups, and one boundary node. See the main text for more information about these nodes. On the left nodes are colored according to their betweenness centrality score [5], and on the right they are colored according to the automatic 3-means clustering of the feature vectors.

the corresponding nodes are at the boundary or at the interface, respectively.

2. *Politics_PR* is the Twitter account of R. Saddler who claims on Nationbuilder.com to be a "social media guy dipping my toe into the liberal side of politics". Still, its node has also many links into the conservative group, which justifies its location at the interface.

3. *DailyCaller* is the Twitter account of the political news website with the same name that has been founded by Tucker Carlson, a political news correspondent for Fox News, and Neil Patel, the former chief policy advisor to Vice President Cheney. The Daily Caller, who claims to reach over 20 million unique readers each month, is not as partisan as the biographies of its founders might suggest. The account has also many links into the liberal group. Hence, it makes sense that this node is classified as interface node.

4. *tironsi75fromNJ* claims on his Twitter account "Bernie Forever". Hence, it seems not justified that this node is considered a boundary node as suggested by our filter and its location in the network layout. Note though that the data set has been collected more than three years ago, and the user's political opinions could have become more articulated since.

5. *NHLabor_news* is the Twitter account of the NH LABOR NEWS blog that is maintained by a group of "proud Union members" from multiple different professions across New Hampshire. Hence, it is obviously correctly located deeply within the liberal group.

6. *TheTeaParty_net* is obviously correctly located deeply within the conservative group.

From our inspection of Twitter accounts present in the TWITTER POLARIZED CROWD network we conclude that boundary nodes typically correspond to nonpolitical Twitter accounts, whereas interface nodes either correspond to nonpolitical accounts or to accounts with a high centrality score and a significant number of followers from both groups, the liberals and the conservatives. Hence, in this network some of the boundary and interface nodes share the same semantics (user type), and need other means like centrality scores to be distinguished.

5.2. Facebook Ego Networks

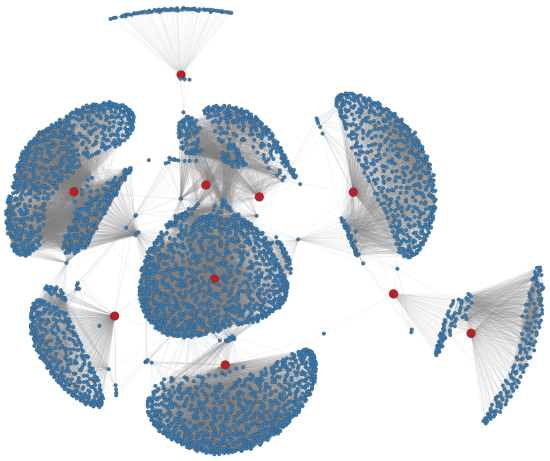
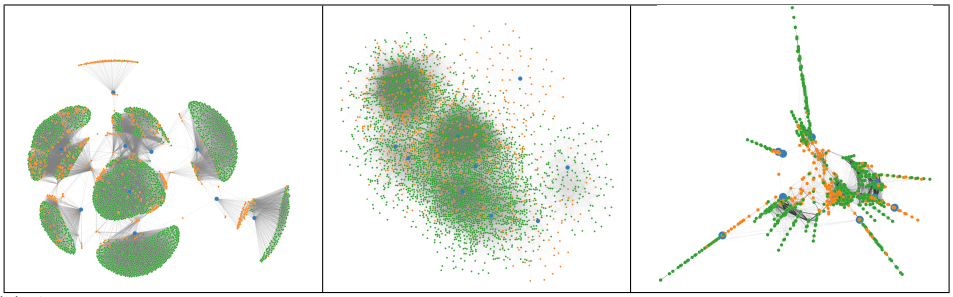


Figure 13: The FACEBOOK EGO NETWORKS data set features the ego networks of ten Stanford students (red nodes) within Facebook. Here again a stress minimization network layout is shown.

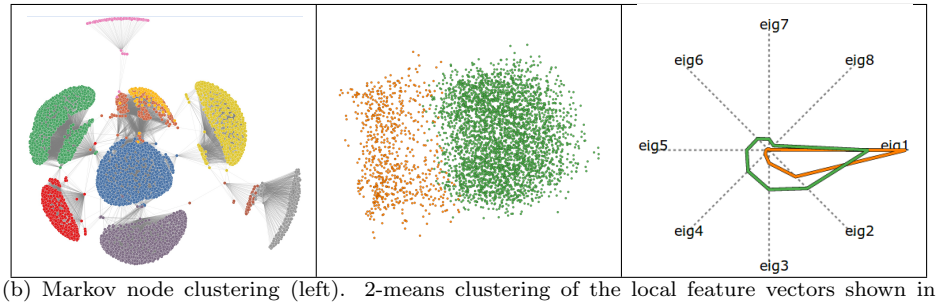
These days it is fairly common for consumers to befriend brands that help them to meet a need or to satisfy a desire. Unfortunately, social network data that are clustered by brand loyalty (think of Nike vs. Adidas) are not available in the public domain. In our second use case we used instead the FACEBOOK EGO NETWORKS data set that has been collected by McAuley and Lescovic [23] for validating an algorithm that automatically detects social circles, like sports teams or relatives, in peoples personal social networks. This network features several densely connected groups that are only loosely coupled and thus also exhibits interface and boundary nodes that we would like to identify. Note though that circles are different from traditional clusters as the membership can be hierarchical. An example for such a hierarchical structure is the circle of college friends that contains the friends from the computer science department which in turn contains the friends under the same advisor. For collecting the data, McAuley and Leskovec have developed a Facebook app that ten participants (Stanford graduate students) used to label the circles in their ego networks. The whole network has $4,039$ nodes and $88,234$ links.

Identifying interface and boundary nodes. Applying a 2-means clustering to the neighborhood feature vectors, see Figure 14 (b) (middle), reveals several loosely coupled groups of nodes (all colored green) and nodes (colored orange) that are mostly found at the interface between the groups in the stress minimization layout. Notably, the groups, some of which can be distinguished already fairly easily from the network layout in Figure 13, do not coincide with the ten personal networks as there are more than ten groups. Markov clustering [13] does a remarkably good job of recovering the ten personal networks, see Figure 14 (b) (left), but does not help in discovering boundary and interface nodes.

Star plots for the two clusters in the set of local feature vectors show that the corresponding neighborhoods are indeed fairly different. The neighborhoods that correspond to nodes within the prominent groups of the network are significantly more spherical than the neighborhoods of the boundary and interface nodes, see Figure 14 (b) (right), which confirms our intuition behind the con-



(a) An automatic 2-means clustering applied to the feature vectors shown directly in the network for different network layouts, namely stress minimization (left), spring embedding (middle), and backbone layout (right). Obviously, boundary and interface nodes are basically impossible to identify in the last two layouts (which holds also true for other networks).



(b) Markov node clustering (left). 2-means clustering of the local feature vectors shown in an MDS plot of these vectors (middle). Star plots that summarize the geometry of the local neighborhood structure for the two clusters (right).

Figure 14: The Clustering analysis on Facebook Ego Networks.

struction of the geometric features.

As for the Twitter Polarized Crowd network, automatically clustering the feature vectors gives together with the stress minimization layout already a good first impression on the location of the interface and boundary nodes, but an interactive exploration of these features brings out finer details, see Figure 15.

A closer look at one of the ego networks. One of the ego networks, see Figure 16 (left), shows a somewhat diffuse substructure in terms of our features. Using our features and tool iteratively on subnetworks enables a closer inspection of such substructures. Relayouting the subnetwork alone, see Figure 16 (right), reveals at least four node clusters A, B, C and D within the subnetwork. Many

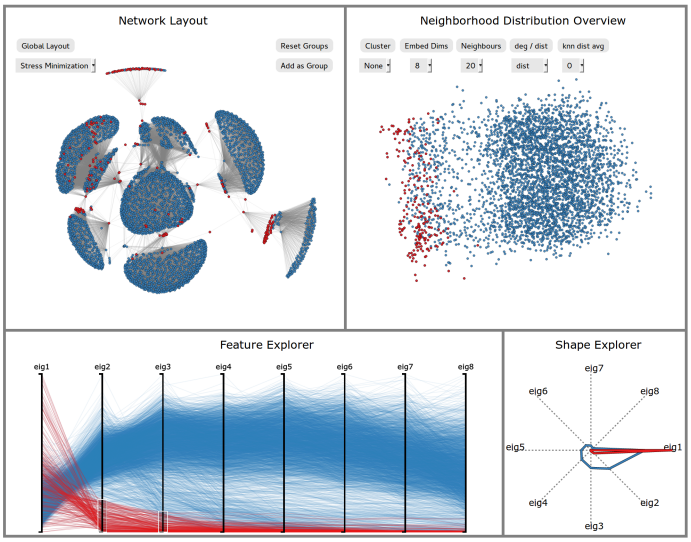


Figure 15: Interactively filtering boundary and interface nodes through the Feature Explorer view. Here only very elongated neighborhoods have been selected as suggested by the apparent clustering structure in the parallel coordinates plot.

of the nodes that have been selected or identified in the 2-means clustering of the feature vectors as boundary and interface nodes are indeed located between these clusters after relayouting the subnetwork. Note that after relayouting almost all of the automatically labeled interface nodes in cluster B have indeed been moved to the interface between the clusters, which demonstrates that it can be difficult to identify interface and boundary nodes only from the global layout, i.e., one would have missed some interface nodes by relying only on the network layout. Note also that cluster D is at the interface to another ego network.

6. Conclusions

We have addressed the problem of identifying interface and boundary nodes in clustered social networks. For instance, clusters might correspond to political parties in political discussion networks, or to subnetworks of social networks that are clustered according to brand loyalty. A common property of boundary

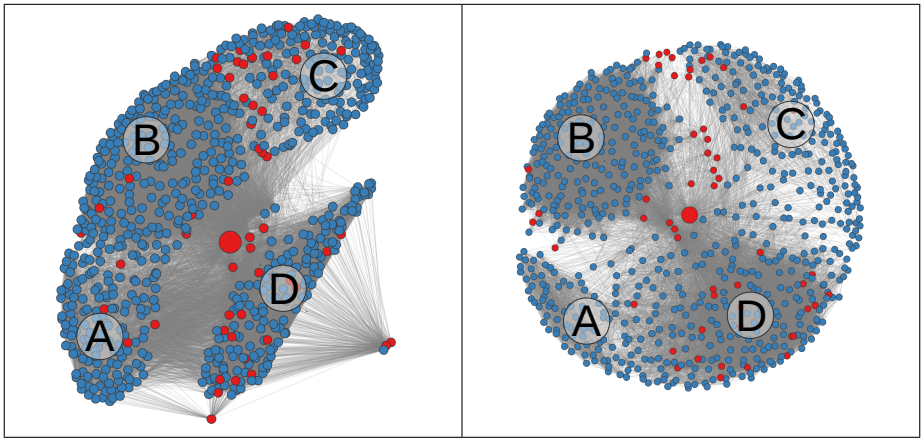


Figure 16: A closer look at one of the ego networks. Original layout on the left, and an individual stress minimization layout for the same subnetwork on the right. The coloring is according to the filtering from Figure 15.

and interface nodes is that they do not belong firmly to any of the clusters. Persons represented by these nodes are thus interesting targets in marketing campaigns that either aim at growing existing clusters or at establishing new clusters (brands).

For the purpose of identifying interface and boundary nodes we have derived geometric features from local network structures. The true potential of these features can only be unlocked by using visual analytics techniques for the analysis of high-dimensional Euclidean point clouds together with classical graph layout strategies. Hence, we have combined several visual techniques and automatic analysis tools like k-means clustering for the exploration of the geometric neighborhood features in a fully linked tool. We have used the tool in two case studies, where the features in conjunction with our interactive approach turned out to be effective in identifying interface and boundary nodes.

References

[1] Eliane Regina de Almeida Valiati, Marcelo Soares Pimenta, and Carla Maria Dal Sasso Freitas. “A taxonomy of tasks for guiding the evaluation

of multidimensional visualizations". In: *Proceedings of the Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV)*. ACM Press, 2006, pp. 1–6.

[2] Giuseppe Di Battista et al. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1998.

[3] Anastasia Bezerianos et al. "GraphDice: A System for Exploring Multivariate Social Networks". In: *Comput. Graph. Forum* 29.3 (2010), pp. 863–872.

[4] Ingwer Borg and Patrick J.E. Groenen. *Modern multidimensional scaling theory and applications*. second. New York: Springer, 2005.

[5] Ulrik Brandes. "A Faster Algorithm for Betweenness Centrality". In: *Journal of Mathematical Sociology* 25 (2001), pp. 163–177.

[6] Shenghui Cheng and Klaus Mueller. "Improving the fidelity of contextual data layouts using a Generalized Barycentric Coordinates framework". In: *Proceedings of IEEE Pacificvis*. Apr. 2015, pp. 295–302.

[7] Shenghui Cheng and Klaus Mueller. "The Data Context Map: Fusing Data and Attributes into a Unified Display". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 121–130. DOI: 10.1109/TVCG.2015.2467552.

[8] Shenghui Cheng, Wei Xu, and Klaus Mueller. "ColorMapND: A Data-Driven Approach and Tool for Mapping Multivariate Data to Color". In: *IEEE Transactions on Visualization and Computer Graphics* 25.2 (2019), pp. 1361–1377. DOI: 10.1109/TVCG.2018.2808489.

[9] Shenghui Cheng, Wei Xu, and Klaus Mueller. "RadViz Deluxe: An Attribute-Aware Display for Multivariate Data". In: *Processes* 5.4 (2017). DOI: 10.3390/pr5040075. URL: https://www.mdpi.com/2227-9717/5/4/75.

[10] Shenghui Cheng et al. "TorusVis$^N D : Unraveling High-Dimensional Torus Networks for Network Traf$ In: *2014 First Workshop on Visual Performance Analysis*. 2014, pp. 9–16. DOI: 10.1109/VPA.2014.7.

[11] Shenghui Cheng et al. “Visualizing the Topology and Data Traffic of Multi-Dimensional Torus Interconnect Networks”. In: *IEEE Access* 6 (2018), pp. 57191–57204. DOI: 10.1109/ACCESS.2018.2872344.

[12] Thomas H. Cormen et al. *Introduction to Algorithms (3. ed.)* MIT Press, 2009.

[13] Stijn van Dongen. “Graph Clustering Via a Discrete Uncoupling Process”. In: *SIAM J. Matrix Analysis Applications* 30.1 (2008), pp. 121–141.

[14] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. “Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation”. In: *IEEE Trans. Vis. Comput. Graph.* 14.6 (2008), pp. 1539–1148.

[15] John A Hartigan. “Printer graphics for clustering”. In: *Journal of Statistical Computation and Simulation* 4.3 (1975), pp. 187–213.

[16] A. Inselberg and B. Dimsdale. “Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry”. In: *Proceedings of IEEE Visualization.* 1990, pp. 361–378.

[17] Eser Kandogan. “Visualizing multi-dimensional clusters, trends, and outliers using star coordinates”. In: *Proceedings of the international conference on Knowledge discovery and data mining (SIGKDD).* 2001, pp. 107–116.

[18] Bongshin Lee et al. “Task taxonomy for graph visualization”. In: *Proceedings of the Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV).* ACM Press, 2006, pp. 1–5.

[19] Stuart P. Lloyd. “Least squares quantization in PCM”. In: *IEEE Trans. Information Theory* 28.2 (1982), pp. 129–136.

[20] Françcois Lorrain and Harrison C. White. “Structural equivalence of individuals in social networks”. In: *The Journal of Mathematical Sociology* 1.1 (1971), pp. 49–80.

[21] Ulrike von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and Computing* 17.4 (2007), pp. 395–416.

[22] Ben Shneiderman Marc A. Smith Lee Rainie and Itai Himelboim. *Mapping Twitter Topic Networks: From Polarized Crowds to Community Clusters*. Available online at Pew Research Center. 2014. URL: http://www.pewinternet.org/.

[23] Julian Mcauley and Jure Leskovec. “Discovering social circles in ego networks”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8.1 (2014), p. 4.

[24] Boaz Nadler et al. “Diffusion maps-a probabilistic interpretation for spectral embedding and clustering algorithms”. In: *Principal manifolds for data visualization and dimension reduction*. Springer, 2008, pp. 238–260.

[25] Takashi Nishikawa and Adilson E Motter. “Discovering network structure beyond communities”. In: *Scientific reports* 1 (2011).

[26] Arlind Nocaj, Mark Ortmann, and Ulrik Brandes. “Untangling hairballs : From 3 to 14 degrees of separation”. In: *Graph Drawing : 22nd International Symposium, GD 2014, Würzburg, Germany, September 24-26, 2014 ; revised selected papers*. Ed. by Christian Duncan ... Lecture Notes in Computer Science 8871. Berlin [u.a.]: Springe, 2014, pp. 101–112. ISBN: 978-3-662-45802-0. DOI: 10.1007/978-3-662-45803-7_9.

[27] Johannes Pretorius, Helen C. Purchase, and John T. Stasko. “Tasks for Multivariate Network Analysis”. In: *Multivariate Network Visualization - Dagstuhl Seminar #13201*. Ed. by Andreas Kerren, Helen C. Purchase, and Matthew O. Ward. Vol. 8380. Lecture Notes in Computer Science. Springer, 2014, pp. 77–95.

[28] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[29] Jianbo Shi and Jitendra Malik. “Normalized cuts and image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905.

[30] Corinna Vehlow, Thomas Reinhardt, and Daniel Weiskopf. “Visualizing Fuzzy Overlapping Communities in Networks.” In: *IEEE Trans. Vis. Comput. Graph.* 19.12 (2013), pp. 2486–2495.

[31] Christophe Viau et al. “The FlowVizMenu and Parallel Scatterplot Matrix: Hybrid Multidimensional Visualizations for Network Exploration”. In: *IEEE Trans. Vis. Comput. Graph.* 16.6 (2010), pp. 1100–1108.

[32] Martin Wattenberg. “Visual exploration of multivariate graphs”. In: *Proceedings of the 2006 Conference on Human Factors in Computing Systems CHI*. 2006, pp. 811–819.

[33] Pak Chung Wong et al. “Graph Signatures for Visual Analytics”. In: *IEEE Trans. Vis. Comput. Graph.* 12.6 (2006), pp. 1399–1413.

[34] Yanhong Wu et al. “Interactive visual summary of major communities in a large network.” In: *PacificVis*. Ed. by Shixia Liu, Gerik Scheuermann, and Shigeo Takahashi. IEEE Computer Society, 2015, pp. 47–54.

[35] Xinyu Zhang, Shenghui Cheng, and Klaus Mueller. “Graphical Enhancements for Effective Exemplar Identification in Contextual Data Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* (2022), pp. 1–1. DOI: 10.1109/TVCG.2022.3170531.