Structural and functional discovery in dynamic networks with non-negative matrix factorization

Shawn Mankad and George Michailidis
Department of Statistics, University of Michigan, Ann Arbor, Michigan 48109-1107, USA
(Received 3 February 2013; revised manuscript received 21 May 2013; published 17 October 2013)

Time series of graphs are increasingly prevalent in modern data and pose unique challenges to visual exploration and pattern extraction. This paper describes the development and application of matrix factorizations for exploration and time-varying community detection in time-evolving graph sequences. The matrix factorization model allows the user to home in on and display interesting, underlying structure and its evolution over time. The methods are scalable to weighted networks with a large number of time points or nodes and can accommodate sudden changes to graph topology. Our techniques are demonstrated with several dynamic graph series from both synthetic and real-world data, including citation and trade networks. These examples illustrate how users can steer the techniques and combine them with existing methods to discover and display meaningful patterns in sizable graphs over many time points.

DOI: 10.1103/PhysRevE.88.042812 PACS number(s): 89.75.Hc

I. INTRODUCTION

Due to advances in data collection technologies, it is becoming increasingly common to study time series of networks. An important research question is how to discover the underlying structure and dynamics in time-varying networked systems. In this work we propose a matrix factorization-based approach for community discovery and visual exploration within potentially weighted and directed network time series. We review and discuss this work in relation to popular approaches for addressing the key problems of community detection and visualization of time series of networks.

There have been many important contributions for community detection in network time series, extensively reviewed in [1,2], from the fields of physics, computer science, and statistics. The basic goal of community detection is to extract groups of nodes that feature relatively dense within-group connectivity and sparser between-group connections [3,4]. A common strategy is to embed the graphs in low-dimensional latent spaces. For instance, Leicht *et al*. [5] use latent variables to capture groups of papers that evolve similarly in citation network data. Sarkar and Moore [6] extend to the dynamic setting a popular latent space model for static data [7] by utilizing smoothness constraints to preserve the coordinates of the nodes in the latent space over time. This article also utilizes a similar low-dimensional embedding strategy. A key difference between this work and [6] is that community membership itself is subject to smoothness conditions in our approach, hence removing the need for a two-stage procedure. This article is also in contrast to previous works that use temporal smoothness constraints for nonoverlapping (hard) community detection [8], estimating time-varying network structure from covariate information [9], predicting network (link) structure [10], or anomaly detection [11,12].

A sequence of non-negative factorizations discovers overlapping community structure, where node participation within each community is quantified and time varying. Other works that consider a single-network cross section have shown advantages of non-negative matrix factorization (NMF) for community detection [13,14]. In addition to a quantification of how strongly each node participates in each community, NMF does not suffer from the drawbacks of modularity optimization methods, such as the resolution limit [15], and can help discover latent structure as shown in the synthetic cell phone network example in this article.

Another advantage of sequential matrix factorization is with large data sets, where data reduction for visualization can help yield additional insights. We use the NMF to transform the time series of networks to a time series for each node, which can be used to create an alternative to graph drawings for visualization of node dynamics. Much of the visualization literature aims to enhance static graph drawing methods with animations that move nodes (vertices) as little as possible between time steps to facilitate readability [16]. However, the reliability of these methods rely on the human ability to perceive and remember changes [17]. Moreover, experiments have discovered that the effectiveness of dynamic layouts is strongly predicted by node speed and target separation [18]. Thus dynamic graph drawings encounter difficulties when faced with a large number of time points, larger graphs that feature abrupt, nonsmooth changes, or if the user is interested in detailed analysis, especially at the individual node level [19,20]. In contrast, static displays facilitate detailed analysis and avoid difficulties associated with animated layouts. This highlights a main advantage our NMF model, namely, creating static displays of node evolutions.

The remainder of this article is organized as follows. We introduce a model for static network data in Sec. II, followed by an extension for dynamic networks in Sec. III. We then test the matrix factorization model on several synthetic and real-world data sets in Sec. IV. In Sec. V we close the article with a brief discussion.

II. NON-NEGATIVE MATRIX FACTORIZATION FOR NETWORK CROSS SECTIONS

The most common factorization is the singular value decomposition (SVD), which has important connections to community detection, graph drawing, and areas of statistics and signal processing [21]. For instance, in classical spectral layout, the coordinates of each node are given by the SVD of graph-related matrices and can be calculated efficiently using algorithms in [22,23]. Recently, there has been extensive interest in spectral clustering [24–26], which aims to discover community structure in eigenvectors of the graph Laplacian

matrix. The method proposed in this paper is similar in spirit, as it also relies on low-rank approximations to adjacency matrices (instead of Laplacian matrices). However, we search for low-rank approximations that satisfy different (relaxed) constraints than orthonormality, namely, that the approximating decompositions are composed of non-negative entries. Such factorizations, referred to as NMF, have been shown to be advantageous for visualization of non-negative data [27–30]. Non-negativity is typically satisfied with networks, as edges commonly correspond to flows, capacity, or binary relationships and hence are non-negative. Non-negative matrix factorization solutions do not have simple expressions in terms of eigenvectors. They can, however, be efficiently computed by formulating the problem as one of penalized optimization and using modern gradient-descent algorithms. Recently, theoretical connections between NMF and important problems in data mining have been developed [31,32] and, accordingly, NMF has been proposed for overlapping community detection on static [13,14] and dynamic [33] networks.

With NMF a given adjacency matrix is approximated with an outer product that is estimated through the minimization

$$\min_{U\geqslant 0, V\geqslant 0} ||A - UV^T||_F^2, \tag{1}$$

where A is the $n \times n$ adjacency matrix and both U and V are $n \times K$ matrices with elements in $\mathbb{R}_+$. The rank or dimension of the approximation K corresponds to the number of communities and is chosen to obtain a good fit to the data while achieving interpretability. An interesting fact about NMF is that the estimates are always rescalable (scale invariant). For example, we can multiply U by some constant c and V by $1/c$ to obtain different U,V estimates without changing their product UV^T. Thus, as seen by the rotational indeterminacy and multiplicative nature of the factorization, NMF is an underconstrained model.

It is, however, straightforward to interpret the estimates due to non-negativity. For instance, $(U)_{ik}(V)_{jk}$ can be interpreted as the contribution of the kth cluster to the edge $(A)_{ij}$. In other words, the expected interaction $(\hat{A})_{ij} = \sum_{k=1}^{K}(U)_{ik}(V)_{jk}$ between nodes i and j is the result of their mutual participation in the same communities [13]. Such an edge decomposition can then be used to assign nodes to communities. For instance, one can proceed by first assigning all edges to the community with largest relative contribution. Then nodes are assigned to communities according to the proportion of their edges that belong to each community. We note that with an NMF-based methodology, the adjacency matrix can be weighted (non-negatively), a potentially appealing feature since many existing analysis tools are arguably only compatible with networks of binary relations.

Though it is not explicitly controlled, standard NMF tends to estimate sparse components. Beyond the additional interpretability that sparsity provides, we find further motivation to encourage sparsity of the NMF estimate when working with networks. For instance, suppose $(A)_{ij} = 0$ for some i,j, that is, there is an absence of an edge between nodes i and j. In the low-rank approximation there is no guarantee that $(\hat{A})_{ij} = 0$, though we expect it to be near zero. A straightforward way to force $(\hat{A})_{ij}$ *exactly* to zero is by anchoring $(U)_{ik} = (V)_{jk} = 0$ for all k and estimating the remaining elements of U and V by the algorithm provided below (see [34] for a similar strategy for multidimensional scaling). However, anchoring is not appropriate with repeated or sequential observations, as an edge can appear and disappear due to noise. Keeping in mind the extension to sequences of networks in the next section, we instead encourage sparsity in the form of an l_1 penalty.

The factorized matrices are obtained through minimizing an objective function that consists of a goodness of fit component and a roughness penalty

$$\min_{U\geqslant 0, V\geqslant 0} ||A - UV^T||_F^2 + \lambda_s \sum_{k=1}^{K} ||V_k||_1, \tag{2}$$

where the parameter $\lambda_s \geqslant 0$. The strength of the penalty is set by the user to steer the analysis, where a larger penalty encourages sparser V. Adding penalties to NMF is a common strategy since they not only improve interpretability, but often improve numerical stability of the estimation by making the NMF optimization less underconstrained. References [35–39] and references therein review important penalized NMF models (see [40–42] for similar approaches with SVD). An advantage of such an approach is that it is easy to modify for particular data sets. For example, a similar l_1 penalty can be included on U if the row space (typically outgoing edges) is of interest.

The estimation algorithm we present is similar to the benchmark algorithm for NMF, known as multiplicative updating [27,28]. The algorithm can be viewed as an adaptive gradient descent. It is relatively simple to implement, but can converge slowly due to its linear rate [43]. In practice we find that after a handful of iterations, the algorithm results in visually meaningful factorizations. The estimation algorithm for the penalized NMF in Eq. (2) is studied in [37,38] and the main derivation steps we present in Table I follow these works.

First, to enforce the non-negativity constraints, we consider the Lagrangian

$$L = ||A - UV^T||_F^2 + \lambda_s \sum_{k=1}^{K} ||V_k||_1 + \mathrm{Tr}(\Phi U^T) + \mathrm{Tr}(\Psi V^T), \tag{3}$$

where Φ, Ψ are Lagrange multipliers. To develop a modern gradient-descent algorithm, we employ the following Karush-Kuhn-Tucker (KKT) optimality conditions, which provide necessary conditions for a local minimum [44]. The KKT optimality conditions are obtained by setting $\frac{\partial L}{\partial U} = \frac{\partial L}{\partial V} = 0$:

$$\Phi = -2AV + 2UV^TV, \tag{4}$$

$$\Psi = -2A^TU + 2VU^TU + 2\lambda_s. \tag{5}$$

TABLE I. Sparse NMF.

1.	Set constant λ_s
2.	Initialize $\{U,V\}$ as dense, positive random matrices
3.	**repeat**
4.	Set $(U)_{ij} \leftarrow (U)_{ij} \frac{(AV)_{ij}}{(UV^TV)_{ij}}$
5.	Set $(V)_{ij} \leftarrow (V)_{ij} \frac{(A^TU)_{ij}}{(VU^TU)_{ij}+\lambda_s}$
6.	**until** Convergence

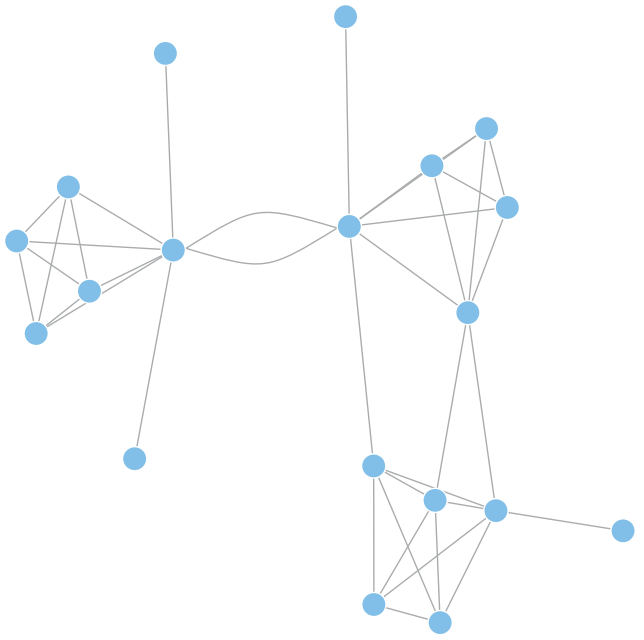

FIG. 1. (Color online) Undirected network with 19 nodes.

Then the KKT complimentary slackness conditions yield

$$0 = (-2AV + 2UV^T V)_{ij}(U)_{ij}, \tag{6}$$

$$0 = (-2A^T U + 2VU^T U + 2\lambda_s)_{ij}(V)_{ij}, \tag{7}$$

which, after some algebraic manipulation, lead to the multiplicative update rules shown in Table I. The algorithm has some notable theoretical properties. Specifically, each iteration of the algorithm will produce estimates that reduce the objective function value, i.e., the estimates improve at each iteration. Minor modifications provided in [45] can be employed to guarantee convergence to a stationary point.

Finally, we note that when the observed graph is undirected, due to symmetry of the adjacency matrix the factorization can be written as

$$A \approx U\Lambda U^T, \tag{8}$$

where Λ is a non-negative diagonal matrix. This is the underlying model investigated in FacetNet [33], with additional constraints on U to satisfy an underlying probabilistic interpretation. The objective function considered in [33] was also based on relative entropy or Kullback-Leibler divergence. We find that such symmetric NMF models are far more sensitive to additional constraints than its general counterpart, especially when dealing with sequences of networks as in the next section. Symmetric NMF has less flexibility since additional constraints strongly influence the reconstruction accuracy of the estimation. In contrast, without imposing symmetry, as V changes, U compensates (and vice versa) in order for the final product to reproduce the data as best as possible. Thus, for tasks of visualization of node evolution and community extraction in dynamic networks, we do not impose symmetry on the factorization.

A. Illustrative examples

1. Community discovery on a toy example

We compare the following methods on a toy example shown in Fig. 1: (i) leading eigenvector- (modularity-) based community discovery [46], (ii) spectral clustering [25], (iii) clique percolation [47] for overlapping community discovery, (iv) classical NMF [Eq. (1)], and (v) sparse NMF [Eq. (2)]. The results of the alternative methodologies are provided in Fig. 2, where we see that even on this toy example, there is disagreement in the recovered community structure. The leading eigenvector solution differs slightly from that of spectral clustering. Taken together, one may suspect that a soft partitioning would result in overlap between the green (upper left) and red (upper right) communities. Yet clique percolation finds overlap between the blue (bottom right) and red (upper right) communities. Classical NMF finds overlap between all three communities, quantifies the amount of overlap (denoted by the pie chart on each node), and decomposes each edge by community. Figure 3 shows that sparse NMF finds a cleaner structure compared to classical NMF. In particular, the sparse NMF solution has less overlap (mixing) between the three groups, while still quantifying the community contribution to nodes and edges.

2. Rank-one factorizations

We show in our experiments (Sec. IV) that a sequence of rank-one matrix factorizations can be the basis for informative displays of time-varying node importance to connectivity. To

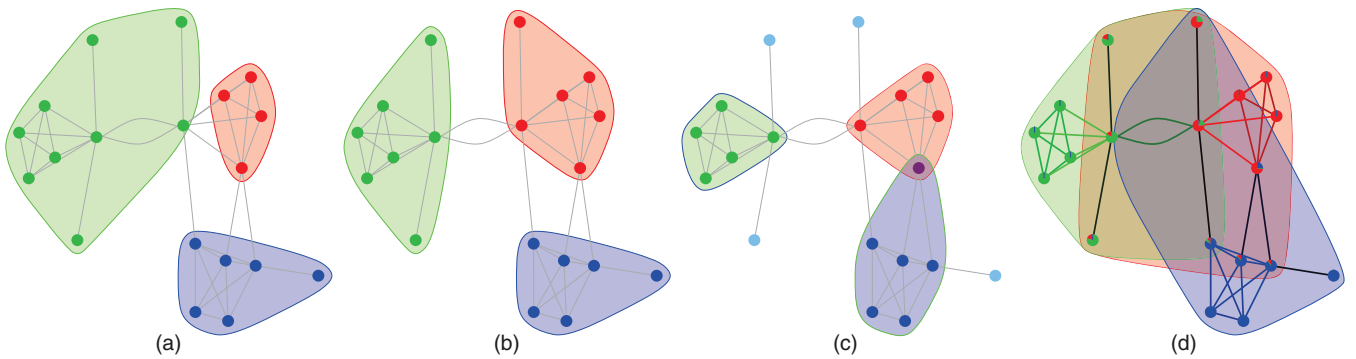

FIG. 2. (Color online) Results using alternative community discovery methods: (a) leading eigenvector, (b) spectral clustering, (c) clique percolation, and (d) classical NMF.

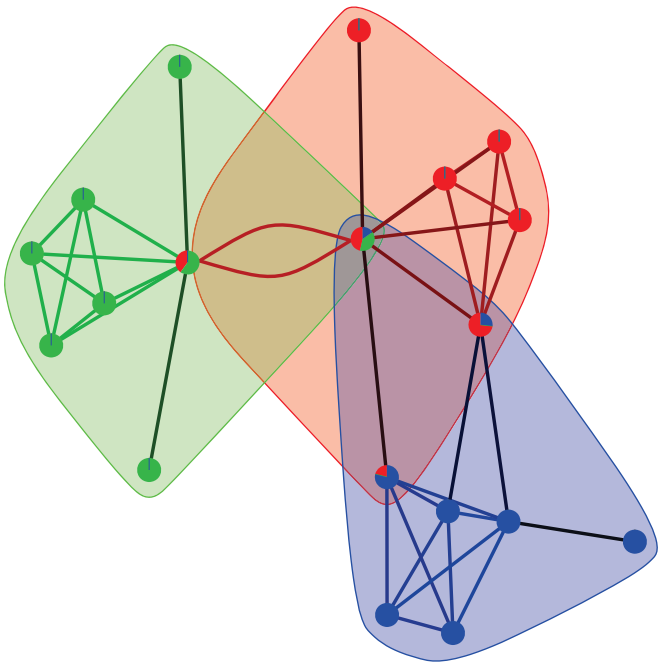

FIG. 3. (Color online) Results from applying sparse NMF (Table I) with $\lambda_s = 5$. Nodes and edges are grouped and colored by their relative contribution of each community.

provide some intuition as to why such a rank-one factorization is a useful basis for visualization and data reduction, consider Fig. 4, which shows graph structures, corresponding NMFs, and Kleinberg's authority and hub scores [48]. Authority and hub scores are computed by the leading eigenvector of A^TA and AA^T, respectively. Subject to rescaling of the NMF estimates, the results are identical. In fact, by the Perron-Frobenius theorem [49], the rank-one NMF solution is always a rescaled version of authority and hub scores. This provides a natural interpretation for the rank-one NMF. For instance, the U vector on the star network highlights the hub node. The V vector show that all peripheral nodes are equal in terms of their authority (incoming connections) and that the central node has no incoming connections. Non-negative matrix factorization vectors of the ring network show each node with an equal score for incoming (authority) and outgoing (hub) connectivity. The fact that U contains larger elements than V is arbitrary. However, the assignment of equal values within U and V shows that each node is equally important to interconnectivity.

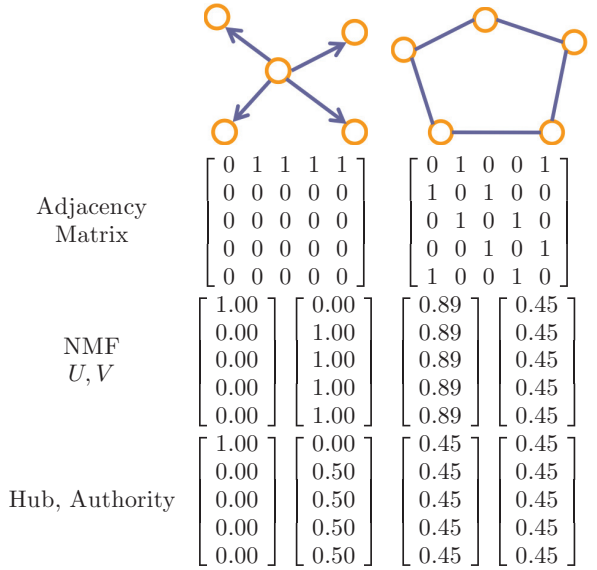

FIG. 4. (Color online) Rank 1 NMF without penalization and Kleinberg's authority and hub scores [48].

III. MODEL FOR DYNAMIC NETWORKS

Given a time series of networks $\{\mathcal{G}_t = (V_t, E_t)\}_{t=1}^T$ with corresponding adjacency matrices $\{A_t\}_{t=1}^T$, the goal is to produce a sequence of low-rank matrix factorizations $\{U_t, V_t\}_{t=1}^T$. To extend the factorization from the previous section to the temporal setting, we impose a smoothness constraint on the basis U_t. This constraint forces the new community structure to be similar to previous time points. Since individual node time series given by U_t are visually smooth, time plots for each node become informative and provide an alternative to graph drawings for visualizing node dynamics. Moreover, time plots are static displays, which facilitate detailed analysis and avoid difficulties associated with animated layouts when given a large number of time points or nodes.

The objective function becomes

$$\min_{\{U_t \geqslant 0, V_t \geqslant 0\}_{t=1}^T} \sum_{t=1}^{T} \left|\left|A_t - U_t V_t^T\right|\right|_F^2 + \lambda_t \sum_{t=1}^{T} \sum_{\tilde{t}=t-\frac{W}{2}}^{t+\frac{W}{2}} ||U_t - U_{\tilde{t}}||_F^2 + \lambda_s \sum_{t=1}^{T} \sum_{k=1}^{K} ||V_{t,k}||_1, \quad (9)$$

where W is a small integer representing a time window. The parameters λ_t, λ_s, and W are set by the user to steer the analysis.

The interpretations of U_t and V_t extend naturally from the previous section, so, for instance, $\sum_k (V_t)_{kj}$ measures the importance of node j (typically corresponding to incoming edges) and $(U_t)_{ik}(V_t)_{jk} / \sum_{k=1}^K (U_t)_{ik}(V_t)_{jk}$ measures the relative contribution of each community to each i, j edge. In principle, the edge decomposition can be used to assign nodes to communities as discussed in the preceding section. However, this approach can be unsatisfactory due to unstable community assignments. As alternative method is to assign communities in terms of U_t, which ensures the stability of the community structure through time, specifically, measuring the contribution of node i to each community with the relative magnitude of the ith element of each dimension of U_t, e.g., $(U_t)_{ik} / \sum_{k=1}^K (U_t)_{ik}$.

We can follow similar steps as in the preceding section to derive a gradient-descent estimation algorithm. First, to enforce the non-negativity constraints, we consider the Lagrangian

$$L = \sum_{t=1}^{T} \left|\left|A_t - U_t V_t^T\right|\right|_F^2 + \lambda_t \sum_{t=1}^{T} \sum_{\tilde{t}=t-W/2}^{t+W/2} ||U_t - U_{\tilde{t}}||_F^2 + \lambda_s \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{j=1}^{K} |V_t(i,j)| + \sum_{t=1}^{T} \mathrm{Tr}\left(\Phi_t U_t^T\right) + \sum_{t=1}^{T} \mathrm{Tr}\left(\Psi_t V_t^T\right), \quad (10)$$

TABLE II. NMF with temporal and sparsity penalties.

1.	Set constants λ_t, λ_s, W
2.	Initialize $\{U_t\}, \{V_t\}$ as dense, positive random matrices
3.	**repeat**
4.	**for** $t = 1, \ldots, T$ **do**
5.	Set $(U_t)_{ij} \leftarrow (U_t)_{ij} \frac{\left(A_t V_t + \lambda_t \sum_{\tilde{t}=t-W/2}^{t-1} U_{\tilde{t}} + \lambda_t \sum_{\tilde{t}=t+1}^{t+W/2} U_{\tilde{t}}\right)_{ij}}{(U_t V_t^T V_t + W\lambda_t U_t)_{ij}}$
6.	Set $(V_t)_{ij} \leftarrow (V_t)_{ij} \frac{(A_t^T U_t)_{ij}}{(V_t U_t^T U_t)_{ij} + \lambda_s}$
7.	**end for**
8.	**until** Convergence

where Φ_t and Ψ_t are Lagrange multipliers.

The following KKT optimality conditions are obtained by setting $\frac{\partial L}{\partial U_t} = \frac{\partial L}{\partial V_t} = 0$:

$$\Phi_t = -2A_t V_t + 2U_t V_t^T V_t - 2\lambda_t \sum_{\tilde{t}=t-W/2}^{t-1} U_{\tilde{t}} - 2\lambda_t \sum_{\tilde{t}=t+1}^{t+W/2} U_{\tilde{t}} + 2W\lambda_t U_t, \tag{11}$$

$$\Psi_t = -2A_t^T U_t + 2V_t U_t^T U_t + 2\lambda_s. \tag{12}$$

Then the KKT complimentary slackness conditions yield

$$0 = \left(-2A_t V_t + 2U_t V_t^T V_t - 2\lambda_t \sum_{\tilde{t}=t-W/2}^{t-1} U_{\tilde{t}} \right)_{ij} (U_t)_{ij} + \left(-2\lambda_t \sum_{\tilde{t}=t+1}^{t+W/2} U_{\tilde{t}} + 2W\lambda_t U_t \right)_{ij} (U_t)_{ij}, \tag{13}$$

$$0 = \left(-2A_t^T U_t + 2V_t U_t^T U_t + 2\lambda_s \right)_{ij} (V_t)_{ij}, \tag{14}$$

which after some algebra leads to the algorithm provided in Table II. The theoretical properties are also the same as in the previous section. Most notably, the estimates of U_t and V_t will improve at each iteration with respect to Eq. (9).

A. Parameter selection

We briefly discuss the important practical matter of choosing K, the inner rank of the matrix factorization. For the goal of clustering, the rank should be equal to the number of underlying groups. The rank can be ascertained by examining the accuracy of the reconstruction as a function of rank.

TABLE III. Cross validation for choosing the number of communities (rank).

1.	Form row holdout set $\mathcal{I}_l \subset \{1, \ldots, n\}$
2.	Form column holdout set $\mathcal{I}_J = \subset \{1, \ldots, n\}$
3.	Set $(\tilde{U}_t, \tilde{V}_t) = \arg\min_{U_t, V_t \geqslant 0} \sum_t \|\|(A_t)_{-\mathcal{I}_l, -\mathcal{I}_J} - U_t V_t^T\|\|_F^2$
4.	Set $\check{U}_t = \arg\min_{U_t \geqslant 0} \sum_t \|\|(A_t)_{\mathcal{I}_l, -\mathcal{I}_J} - U_t \tilde{V}_t^T\|\|_F^2$
5.	Set $\check{V}_t = \arg\min_{V_t \geqslant 0} \sum_t \|\|(A_t)_{-\mathcal{I}_l, \mathcal{I}_J} - \tilde{U}_t V_t^T\|\|_F^2$
6.	Set $(\hat{A}_t)_{\mathcal{I}_l, \mathcal{I}_J} = \check{U}_t \check{V}_t^T$
7.	Compute test error $\sum_t \|\|(A_t)_{\mathcal{I}_l, \mathcal{I}_J} - (\hat{A}_t)_{\mathcal{I}_l, \mathcal{I}_J}\|\|_F^2$

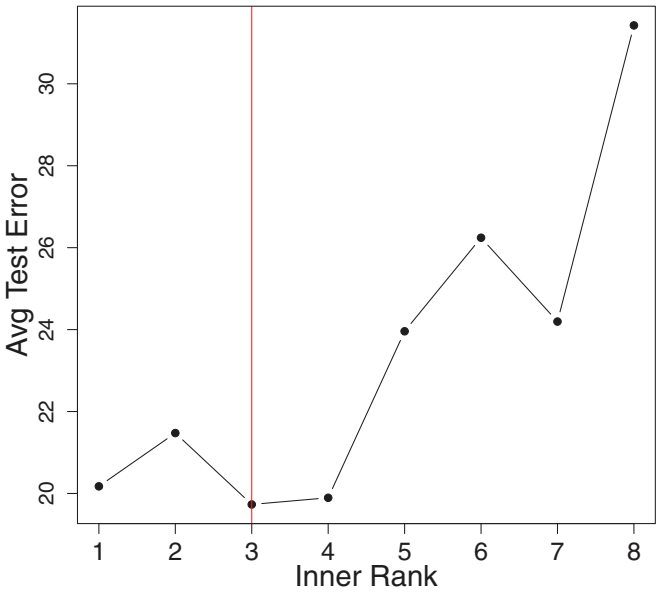

FIG. 5. (Color online) Cross validation indicates three communities (rank 3) featuring the lowest average test error for the toy example.

However, this tends to rely on subjective judgments and overfit the given data. Cross-validation-based approaches are theoretically preferable and follow the same intuition.

The idea behind cross validation is to use random subsets of the data from each network cross section to fit the model and another subset to assess accuracy. Different values of K are then cycled over and the one that corresponds to the lowest test error is chosen.

Due to the data structure, we employ two-dimensional cross validation. Two dimensional refers to the selection of *submatrices* for our training and test data. Special care is taken to ensure that the same rows and columns are held out of every

FIG. 6. (Color online) Cell phone network from a day using a force-directed layout algorithm in igraph. Node 200 is colored black.

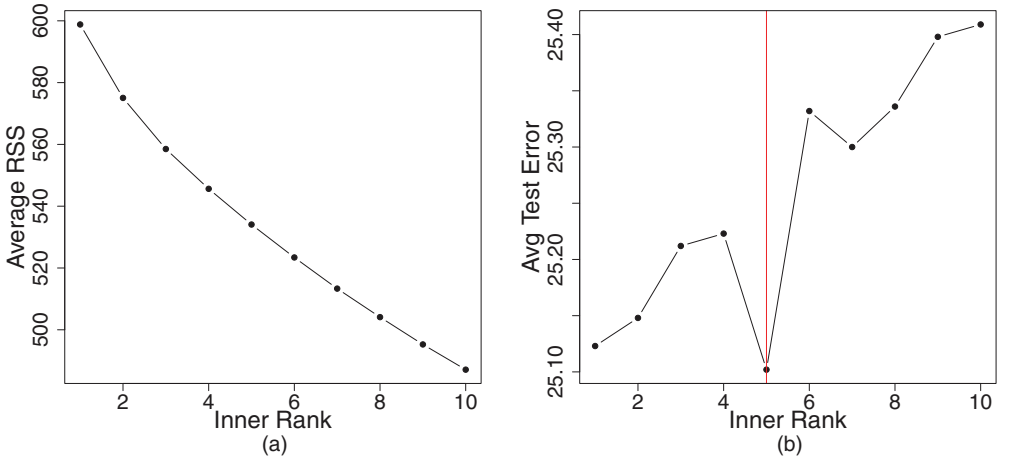

FIG. 7. (Color online) Choosing K for the Catalano communications network. (a) Average residual sum of squares. (b) Average test error obtained via cross validation [(5×5)-fold] for different approximation ranks. Cross validation indicates five communities as optimal.

adjacency matrix and the dimensions of the training and test sets are identical.

The holdout pattern divides the rows into k groups, the columns into l groups, and then uses the corresponding kl submatrices to fit and test the model. In each submatrix, the given row and column group identifies a held out submatrix that is used as test data, while the remaining cells are used for training. The algorithm is shown in Table III. The notation in the algorithm uses $\mathcal{I}_l$ and $\mathcal{I}_J$ as index sets to identify submatrices in the each data matrix.

We then cycle over different values of K to choose the one that minimizes average test error. Figure 5 shows that this procedure correctly identifies three communities for the toy example. Consistency results are

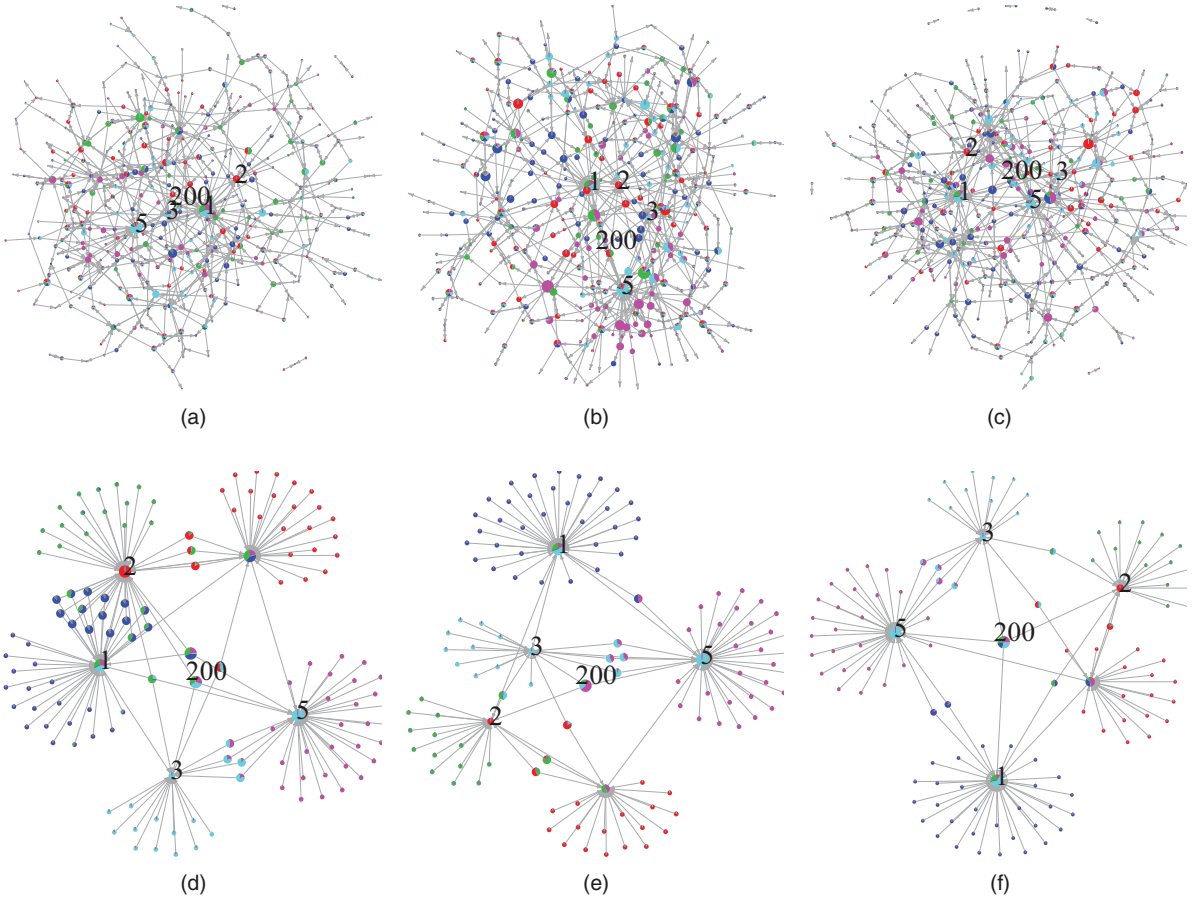

FIG. 8. (Color online) From left to right, the columns are organized by day: (a) and (d) day 5, (b) and (e) day 6, and (c) and (f) day 7. (a)–(c) Raw Catalano networks colored by the U_t community structure and (d)–(f) filtered Catalano networks colored by the U_t.

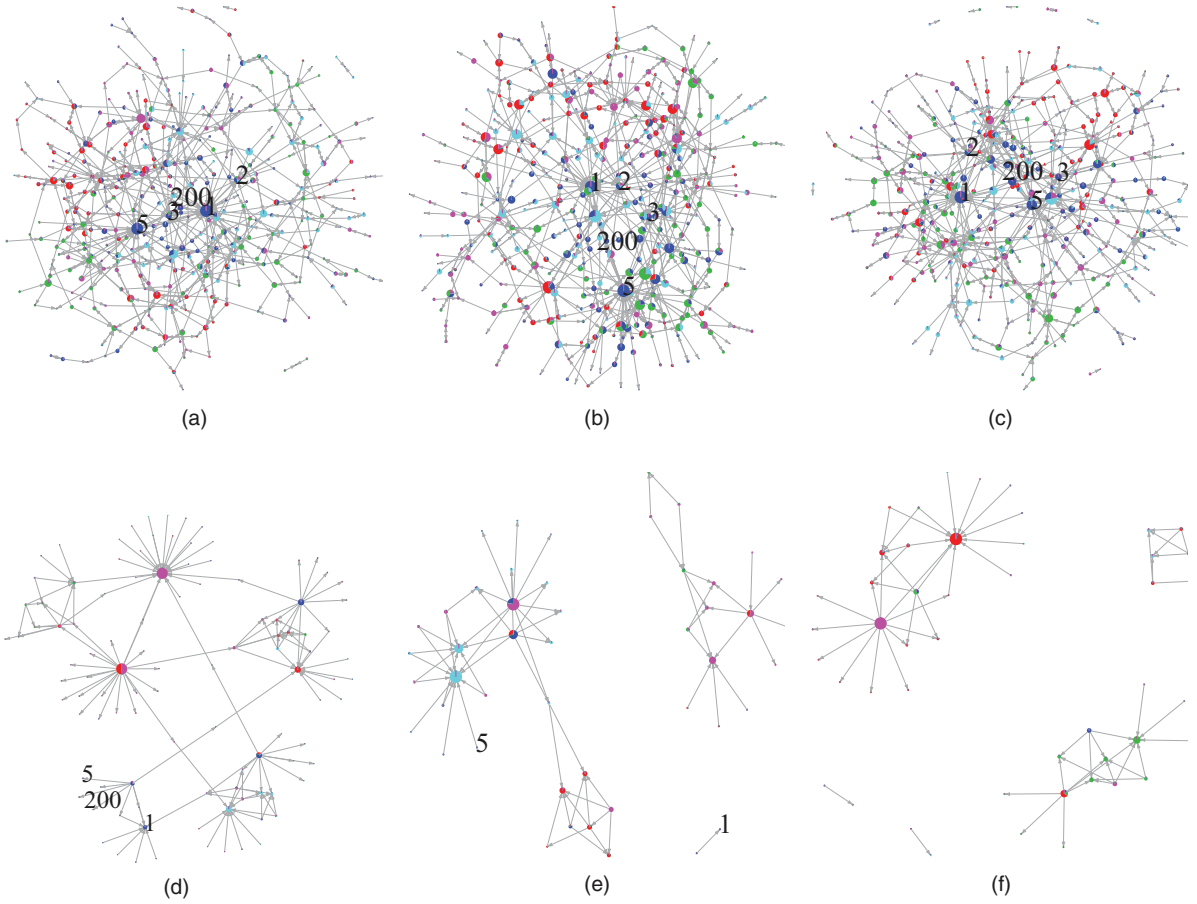

FIG. 9. (Color online) From left to right, the columns are organized by day: (a) and (d) day 5, (b) and (e) day 6, and (c) and (f) day 7. (a)–(c) Raw Catalano networks colored by the FacetNet factorization and (d)–(f) corresponding FacetNet-filtered Catalano networks.

developed in [50] to provide theoretical foundations for this approach.

In principle the cross-validation procedure can be used to select the penalties λ_t and λ_s and the time window W. However, considering the scale of many modern network data sets, this would require too much computing time. Instead we typically choose the penalties by hand to emphasize readability and interpretability of the results, keeping in mind that if either penalty is set too large then the estimation results in degenerate solutions. For instance, the algorithm suffers from numerical instabilities when λ_s is too large since all V_t elements are zero. If λ_t is set to an extremely large number, then U_t will be approximately constant for all time periods, so the effective model is $A_t \approx U V_t^T$, e.g., the community structure is fixed for all observations.

The parameter W controls the number of neighboring time steps to locally average. Larger values of W mean that the model has more memory so it incorporates more time points for estimation. This risks missing sharper changes in the data and only detecting the most persistent patterns. In contrast, small values of W make the fitting more sensitive to sharp changes, but increase short-term fluctuations due to smaller number of observations. We set $W = 2$ (looking one time period ahead and before) for all presented experiments. Larger values could be used in very noisy settings to further smooth results.

IV. EXPERIMENTS

In this section we test the model on both synthetic and real-world examples. The synthetic networks allow us to validate the model's ability to highlight known community structure and node evolution, while the real examples exhibit the model's performance under practical conditions.

A. Synthetic networks

1. Catalano communication network

The first example utilizes the Catalano social network, which was part of the Visual Analytics Science and Technology (VAST) 2008 challenge [51]. The synthetic data consist of 400 unique cell phone identifications over a ten day period. Altogether, there are 9834 phone records with the following fields: calling phone identifier, receiving phone identifier, date, time of day, call duration, and cell tower closest to the call origin. The purpose of the challenge was to characterize the social structure over time for a fictitious, controversial sociopolitical movement. In particular, the challenge requires identifying five key individuals that organize activities and communications for the network; a hint was given to challenge participants that node 200 is one of the persons of interest.

We use the first seven days of data to illustrate our methodology since there is a strong change in the connection

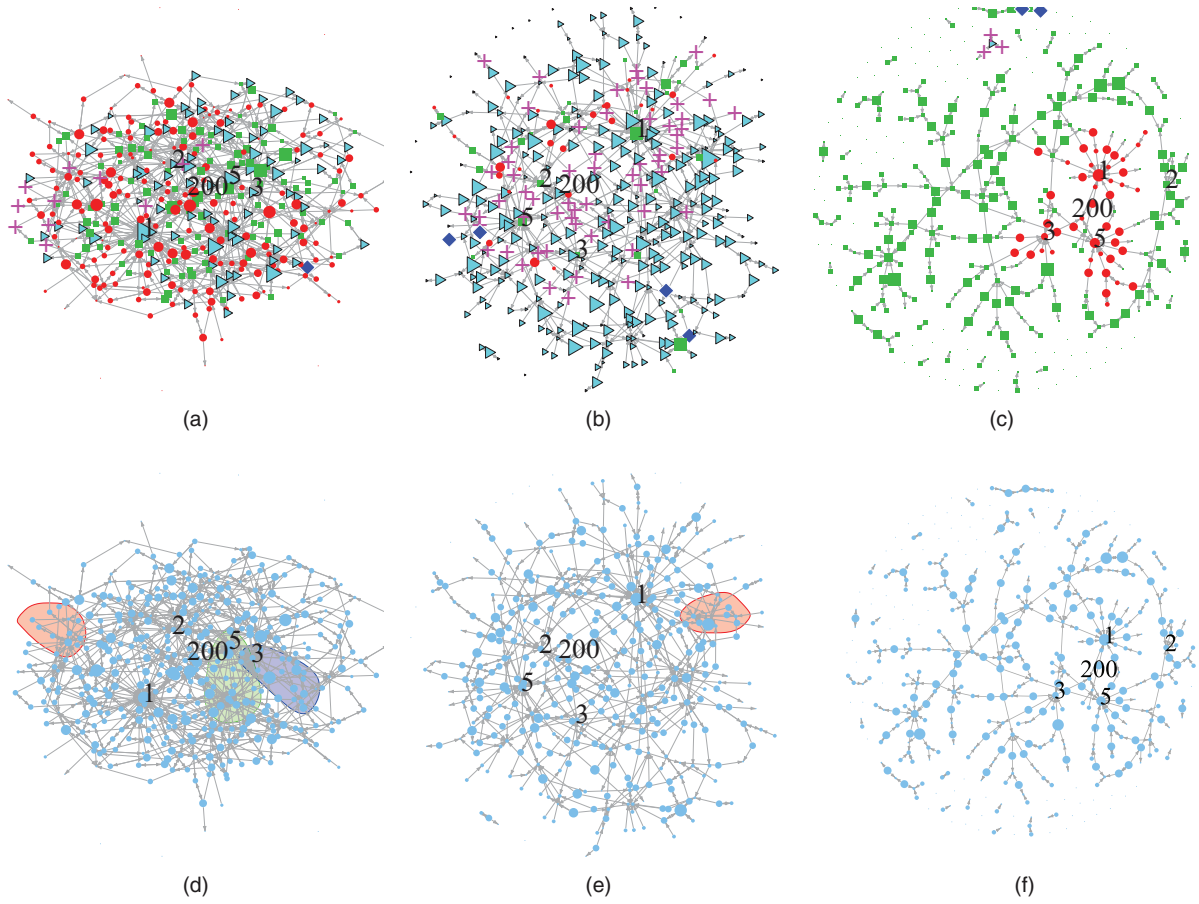

FIG. 10. (Color online) From left to right, the columns are organized by threshold: (a) and (d) threshold equal to 2, (b) and (e) threshold equal to 3, and (c) and (f) threshold equal to 4. (a)–(c) Spectral clustering is applied to the collapsed data (averaged over time) and (d) and (e) clique percolation is applied to the collapsed data.

patterns from day 8 to 10 for node 200 (see [51,52] and references therein). Directed networks are constructed daily by drawing an edge from the caller to the receiver. Figure 6 shows an example of one day's network. The graph is too cluttered to visually identify leaders of the network or get a sense of the network structure.

We fit a sequence of rank-5 NMFs, as identified in Fig. 7 through cross validation, with a large temporal penalty to highlight the most persistent interactions, thus discovering latent structure. Figure 8 shows two sets of graph drawings for three days (due to space limitations), with the nodes decomposed according to their community membership. The first row shows the graph constructed directly from the data, while the second row shows graph drawings of the *fitted* model $\hat{A}_t = U_t V_t^T$. The clustering results applied to the raw data are not interpretable, as the data are simply too cluttered. However, the persons of interest and the hierarchical structure of the communication network are clearly shown when considering the fitted networks. One can visually identify that node 200 consistently relays information to its neighbors (1, 2, 3, and 5), who disseminate information to their respective subordinates. We can also see that nodes higher up on the organizational hierarchy tend to belong to multiple communities, presumably since they disseminate information to different groups of subordinates.

Figure 9 shows the results of applying FacetNet [33], an alternative NMF methodology for dynamic overlapping community detection. FacetNet applies an underlying model with less flexibility resulting in poor reconstructions of the data, as seen in the fitted networks. We also collapse the data into a single-network snapshot in order to apply static clustering algorithms (Fig. 10). First, an edge is kept only if it was observed more than *threshold* days. Then spectral clustering and clique percolation are applied to the resultant network snapshot. All alternative methods struggle, as the data are too hairball-like. In contrast, the fitted penalized NMF model provides a unified framework to filter the network and visualize community structure. VAST never officially released correct answers for the challenge. However, our analysis closely matches winning entries [52–54]. Treating the conclusions of the entries as the ground truth, we have provided a simple workflow that uncovers patterns in the data that are not directly obtainable with traditional methods.

2. *Preferential attachment process*

In this simulation, nodes attach according to a preferential attachment model [55,56] until 10 000 nodes have attached to the embedding. We observe this growing process at 100 uniformly spaced time points. Thus, at each time point 100 new nodes attach to the graph. We use source code from a network

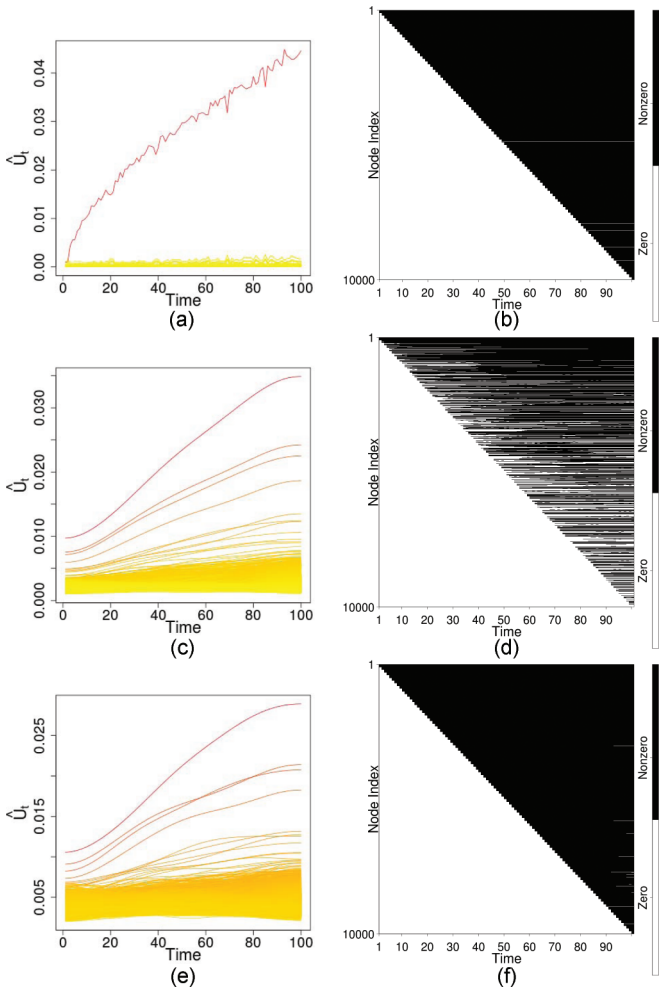

FIG. 11. (Color online) Fitted values for U_t and V_t over time for the preferential attachment simulation. In the time plots of U_t (left column), each line corresponds to a node on the graph. In the binary heat maps of V_t (right column), each row corresponds to a node on the graph and time varies along the horizontal axis. Wherever a row is colored black, the corresponding paper had nonzero coordinates within V_t. From top to bottom, each row displays increasing levels of penalization from (a) and (b) no penalties to (c) and (d) $\lambda_t = 50, \lambda_s = 5$ and (e) and (f) $\lambda_t = 100, \lambda_s = 5$.

MATLAB toolbox [57] that generates preferential attachment graphs according to the standard model.

In the preferential attachment model, $\Pi(i)$, which represents the probability that a new node connects to node i, depends on node i's degree. Specifically, we have

$$\Pi(i) \propto d_i, \tag{15}$$

where d_i is the degree of the ith node. This generating framework leads to networks whose asymptotic degree distribution follows a power-law distribution with parameter $\gamma = 3$. Graphs with heavy-tailed degree distributions are commonly observed in a variety of areas such as the Internet, protein interactions, and citation networks, among others [58].

In practice, an analyst would not know that the data comes from a preferential attachment process, in which case an exploratory analysis may include inspecting the network sequence on a set of standard metrics (degree, transitivity,

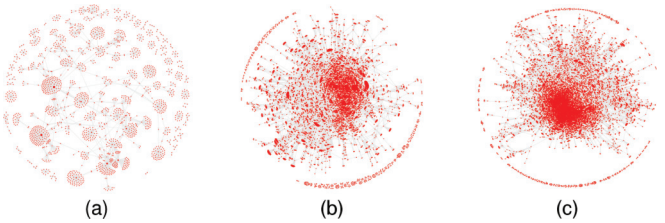

FIG. 12. (Color online) Graph layouts of the arXiv data from (a) January 1995, (b) January 1998, and (c) January 2000. Due to the size of the networks, it quickly becomes difficult to discern paper (node) properties.

centrality, etc.), graph drawings, as well as community detection approaches. We believe that a sequence of one-dimensional ($K = 1$) penalized NMFs can serve as the basis for a complimentary exploratory tool that helps uncover different connectivity patterns and evolution in the data. In particular, due to the smoothness penalty, time plots in U_t for each node become useful for uncovering the number and types of node evolutions in the data. Similarly, binary heat maps or displays of the sparsity pattern of V_t are useful to identify when nodes or groups become significantly active.

Since preferential attachment networks have been extensively studied, we show only the NMF-based displays. Figure 11 shows important (hub) nodes with distinct trajectories that indicate their increasing importance to the network over time. The V_t sparsity features a pseudo-upper triangular form. This corresponds to the node attachment order and reflects that nodes permanently attach after connecting to the network. Such displays can be created quickly and can help the

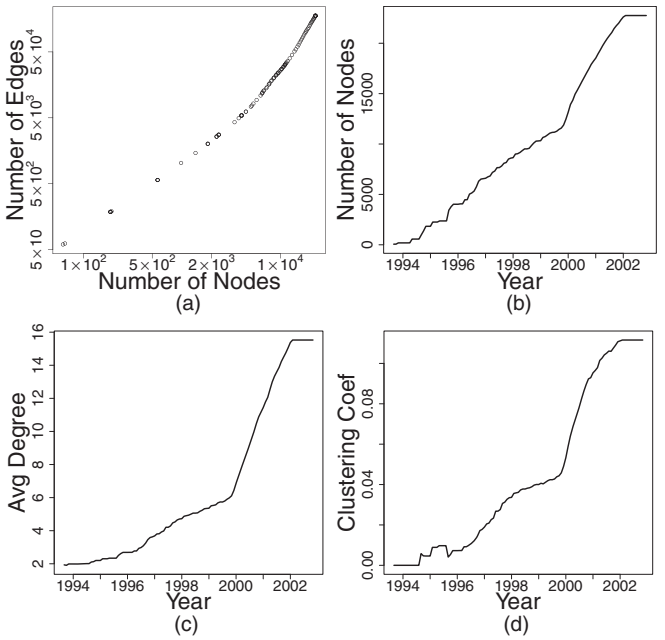

FIG. 13. arXiv network statistics over time. The kink near January 2000 indicates sudden, rapid growth. (a) A scatter plot on a log-log scale of the number of nodes and edges for each observed citation network. (b) Time series of number of nodes in each network. (c) Time series of average degree in each network. (d) Time series of clustering coefficient for each network.

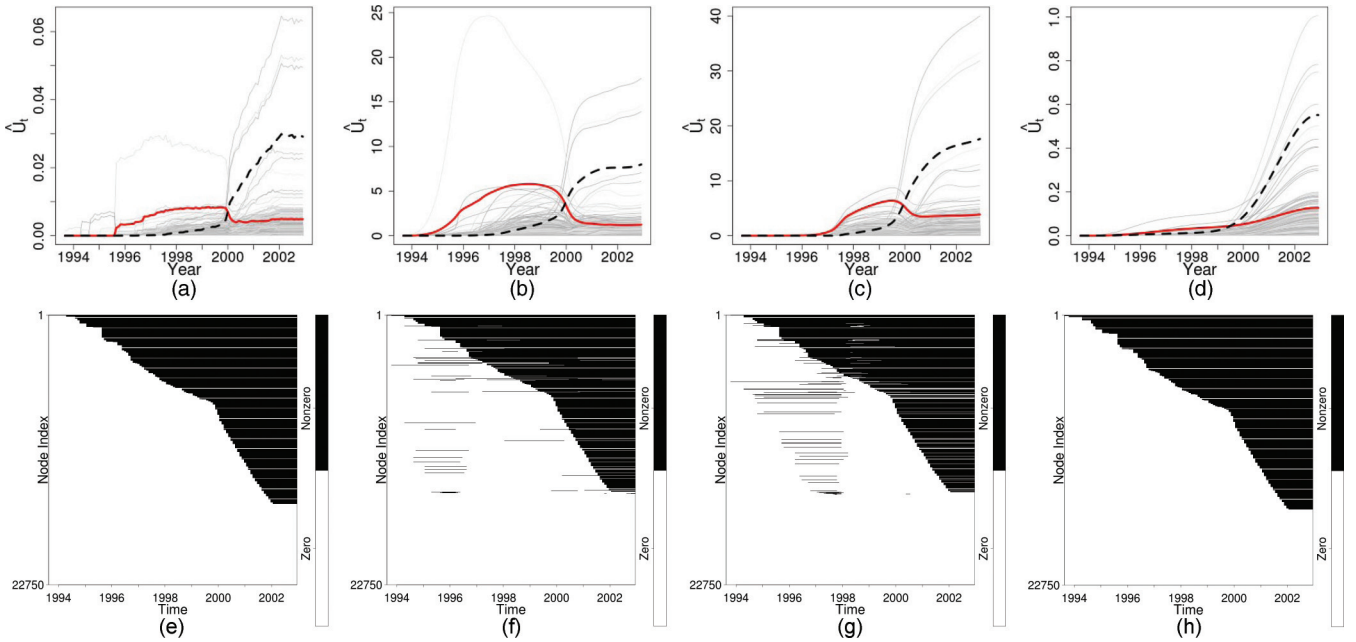

FIG. 14. (Color online) Fitted values for U_t and V_t for the arXiv data. From left to right, each column shows (a) and (e) $\lambda_t = \lambda_s = 0$, (b) and (f) $\lambda_t = 5, \lambda_s = 1$, (c) and (g) $\lambda_t = 5, \lambda_s = 5$, and (d) and (h) $\lambda_t = 5, \lambda_s = 10$. In the time plots of U_t (top row), each light gray line corresponds to a paper (node) on the graph. The bold lines show the average of the ten papers with highest average $\hat{U}$ from 1996 to 1999, and 2000 onward (dashed). Each row in the binary heat maps of V_t (bottom row) corresponds to a paper and time varies along the horizontal axis. Wherever a row is colored black, the corresponding paper had nonzero coordinates within V_t.

process of identifying interesting nodes, formulating research questions, and so on.

Also shown in Fig. 11 is that penalization is important to the usefulness and interpretability of the displays. For instance, without a temporal penalty, the time plots emphasize only the highest degree node. With appropriate penalties, an analyst can visually identify the different hub nodes and their evolution.

B. Real networks

1. *arXiv citations*

We investigate a time series of citation networks provided as part of the 2003 KDD Cup [59]. (The graphs are from the e-print service arXiv for the "high energy physics theory" section.)

The data cover papers in the period from October 1993 to December 2002 and are organized into monthly networks. In particular, if paper i cites paper j, then the graph contains a directed edge from i to j. Any citations to or from papers outside the data set are not included. Following convention, edges are aggregated, that is, the citation graph for a given month will contain all citations from the beginning of the data up to and including the current month. Altogether, there are 22 750 nodes (papers) with 176 602 edges (references) over 112 months.

As a first step towards investigating the data, we draw the network at different points in time in Fig. 12. Even when considering a single time point, it quickly becomes difficult to discern paper (node) properties due to the large network size. Thus the data require network statistics and other methods to

TABLE IV. Top ten papers with highest average $\hat{U}$ from 1996 to 1999. The number of citations counts all references to the work, including by papers outside our data. These counts were obtained via Google.

Title	Authors	In degree	Out degree	No. of citations
Heterotic and Type I String Dynamics from Eleven Dimensions	Horava and Witten	783	18	2334
Five-branes and M-Theory On An Orbifold	Witten	169	15	251
Type IIB Superstrings, BPS Monopoles, and Three-Dimensional Gauge Dynamics	Hanany and Witten	437	20	844
D-Branes and Topological Field Theories	Bershadsky *et al.*	271	15	463
Lectures on Superstring and M Theory Dualities	Schwarz	247	68	534
D-Strings on D-Manifolds	Bershadsky *et al.*	172	22	247
String Theory Dynamics in Various Dimensions	Witten	263	0	2263
Branes, Fluxes and Duality in M(atrix)-Theory	Ganor *et al.*	184	16	243
Dirichlet-Branes and Ramond-Ramond Charges	Polchinski	370	0	2592
Matrix Description of M-theory on T^5 and T^5/Z_2	Seiberg	208	30	353

TABLE V. Top ten papers with highest average $\hat{U}$ from 2000 onward. The number of citations counts all references to the work, including by papers outside our data. These counts were obtained via Google.

Title	Authors	In degree	Out degree	No. of citations
The Large N Limit of Superconformal Field Theories and Supergravity	Maldacena	1059	2	10697
Anti De Sitter Space And Holography	Witten	766	2	6956
Gauge Theory Correlators from Non-Critical String Theory	Gubser *et al.*	708	0	6004
String Theory and Noncommutative Geometry	Seiberg and Witten	796	12	3833
Large N Field Theories, String Theory and Gravity	Aharony *et al.*	446	74	3354
An Alternative to Compactification	Randall and Sundrum	733	0	5693
Noncommutative Geometry and Matrix Theory: Compactification on Tori	Connes *et al.*	512	3	1810
M Theory As A Matrix Model: A Conjecture	Banks *et al.*	414	0	2460
D-branes and the Noncommutative Torus	Douglas and Hull	296	2	866
Dirichlet-Branes and Ramond-Ramond Charges	Polchinski	370	0	2592

extract the structure and infer the dynamics in the network sequence. Network statistics, shown in Fig. 13, provide some additional insight. There is a noticeable increase in network growth around the year 2000, which is commonly attributed to papers that reference other works before the start of the observation period (see [60]). As we move away from the beginning of the data, papers primarily reference other papers belonging to the data set. Additional statistical properties of the data were discussed in [60], which found that the networks feature decreasing diameter over time and heavy-tailed degree distributions.

To visualize how nodes in the network evolved, Fig. 14 displays results from the matrix factorization model using a sequence of one-dimensional approximations ($K = 1$). The adjacency matrix is constructed so that U_t scores nodes by their importance to the average *incoming* connections and $(U_t)_{1j}$ measures the time-varying authority of paper j; V_t yields similar scores based on outgoing connections. As observed with the preferential attachment experiment, the paper trajectories are smoothed effectively and important dynamics are highlighted by employing penalties. Specifically, there are two important periods in the data. The first period covers 1996–1999 and features papers mostly on an extension of string theory called M-theory. M-theory was first proposed in 1995 and led to new research in theoretical physics. A number of scientists, including Witten, Sen, and Polchinski, were important to the historical development of the theory and as seen in Tables IV and V, our NMF approach identifies these important authors and their works. From 1999 to 2000 the rate of citations to these papers tended to decrease, while focus shifted to other topics and subfields that M-theory gave rise to. These citation patterns are reflected in the bold and dashed trajectories in Fig. 14. The displays of V_t sparsity show that papers do not appear uniformly throughout time. Instead, as other network statistics show, papers attach at a faster rate around year 2000.

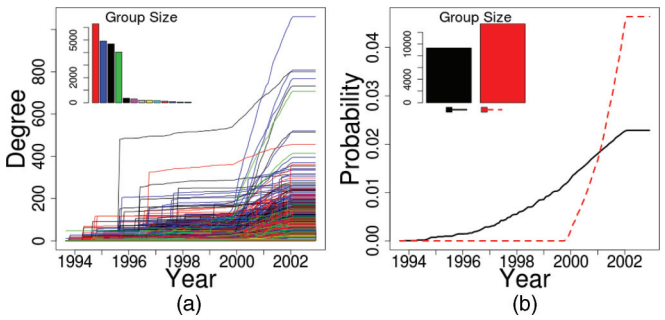

FIG. 15. (Color online) (a) Degree of each node over all time points, shaded and colored by the leading eigenvector groupings. (b) Time profiles for each group based on the mixture model of [5]. Specifically, (b) shows the estimated probability that a particular citation received by a document in the group is made in each month.

We provide comparisons with the alternative methodologies utilized in [5] to investigate dynamic citation network from the US Supreme Court. First, we apply the leading eigenvector modularity-based method for community discovery [46] to the fully formed citation network ($t = 112$). The second alternative methodology is a mixture model in [5] to extract groups of papers according to their common temporal citation profiles.

Figure 15(a) shows the degree of each paper over time, shaded and colored by the leading eigenvector community assignments. The optimal number of groups is over 200. There are four large groups of papers, with the other groups containing only a handful of papers. This approach does not utilize the temporal profile of each paper and as a consequence the groups are interpretable from a static connectivity point of view only.

Figure 15(b) shows reasonable time profiles from the mixture model. One group grows slowly from the beginning of the observational period, while the other group experiences rapid growth starting around the year 2000. These results complement the NMF-based Fig. 14 and together provide a robust methodology to identify important papers, as well as characterize the data in terms of the number and types of different nodes or groups in the data.

2. *Global trade flows*

In this example, we analyze annual bilateral trade flows between 164 countries from 1980 to 1997 [61]. Thus we observe a dynamic, weighted graph at 18 time points, where each directional edge denotes the total value of exports from

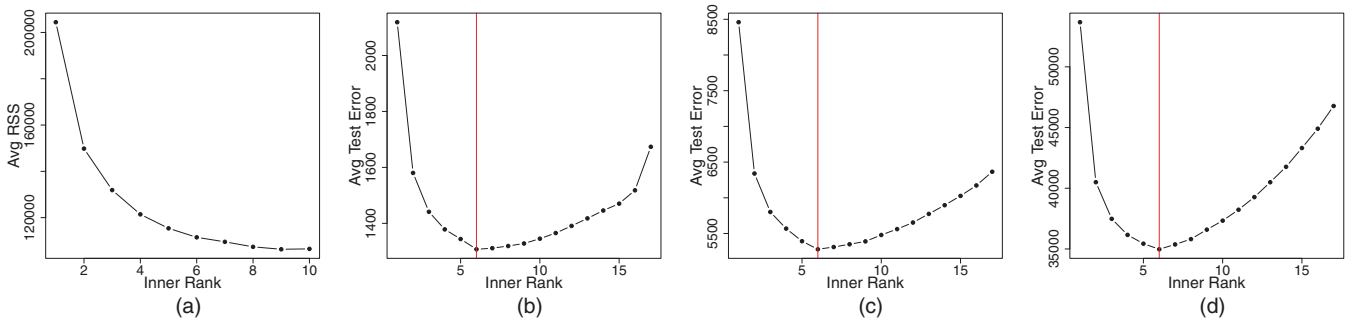

FIG. 16. (Color online) Choosing K for world trade data. (a) Average residual sum of squares. The average test error obtained via cross validation with (b) two partitions, (c) five partitions, and (d) ten partitions. Cross validation consistently indicates six communities ($K = 6$) as optimal.

one country to another. Since trade flows can differ in size by orders of magnitude, we work with trade values that are expressed in log dollars.

We fit a sequence of rank-6 NMFs, as identified in Fig. 16 through cross validation, and display the network based on fitted trade flows ($\hat{A}_t = U_t V_t^T$) in Fig. 17. We show only three years (1980, 1990, and 1997) due to space constraints.

All countries belong to more than one community, which reflects the interconnected nature of the global economy. However, there are countries, primarily from Africa and Central America, that are dominated by a single community or belong to only a subset of the six communities. For instance, in 1997, many Central American countries [circled in Fig. 17(c)] connect only with the USA and hence belong mostly to a single community.

There are also interesting findings that correspond with historical events. For instance, in 1980 there is a strong community [circled in Fig. 17(a)] consisting of countries aligned with the former USSR, which acted as a hub. However, by 1990, this community has dissolved and is reflected in the edge and node colorings of these countries (more diversified trading relationships). In 1990, we also see the emergence of the so-called Asian miracles, countries in Asia that experienced persistent and rapid economic growth in the 1990s [62,63]. These countries move closer to the center of the trading network with membership in all communities.

V. DISCUSSION

The main idea behind the approach presented in this paper is to abstract the network sequence to a sequence of lower-dimensional spaces using matrix factorizations for visual exploration, community detection, and structural discovery. Next we highlight some of the strengths and weaknesses of this approach.

A. Strengths

An important benefit is the versatility and scalability of matrix factorization model. Table VI shows run-times for all experiments. The computational cost is low enough to use in combination with other analysis and visualization tools. Moreover, the penalized NMF approach is compatible with both binary and weighted networks.

Using the model as a basis for an exploratory visual tool can help with data reduction and discovery of latent evolving structure. The estimates of U_t and V_t can be used for community discovery or a ranking of nodes based on their importance to connectivity for subsequent analysis. Displays of the factorizations can provide a sense of the data complexity, namely, the types and number of node evolutions.

B. Weaknesses

The optimal choice of tuning parameters (λ_t, λ_s) is dependent on perception and how the edge weights are scaled.This

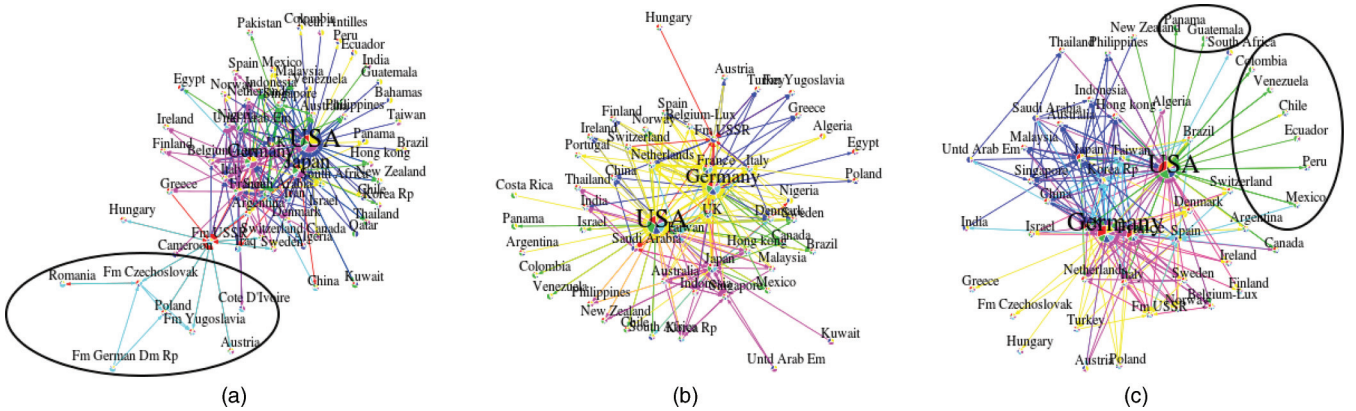

FIG. 17. (Color online) World trade networks, where countries and their edges are shaded and colored corresponding to their membership in six communities. The trading network is shown from (a) 1980, (b) 1990, and (c) 1997.

TABLE VI. Average run times for the penalized NMF with temporal and sparsity penalties. The computational time scales approximately linearly with the number of time points and nodes.

Data	Nodes	Time points	Run time (s)
Catalano	400	7	0.29
world trade	164	18	0.51
preferential attachment	10000	100	39.45
arXiv citations	22750	112	60.64

can limit the benefits of the proposed approach when given multiple data sets.

Time plots and heat maps to visualize each factor yield limited information about global topology. For example, one can see from Figs. 11 and 14 that there are dominant nodes, but, in principle, there could be many topologies that feature dominant nodes. One cannot say for sure without additional analysis that the networks follow a particular connectivity model. Thus, combining the matrix factorization model in this article with existing analysis and visualization tools can provide a more comprehensive analysis of the data [64].

C. Future work

An important area of exploration would be to systematically compare penalized versions of NMF and SVD. In this work we chose to focus on NMF since we find the corresponding displays preferable in terms of interpretability. This is generally consistent with existing literature on matrix factorization. However, SVD of graph-related matrices has deep connections to classical spectral layout and problems in community detection. There may be classes of graph topologies and particular visualization goals under which SVD is preferable.

There could also be other types and combinations of penalties that are useful in visualization and detection of graph structure. For instance, depending on the precise meaning of a directional edge, one may desire both smoothness and sparsity for U_t, V_t, or both factors. Nonetheless, variants on the penalty structure will result in models that require roughly the same computational costs. Thus this work provides evidence that penalized matrix factorization models are promising for structural and functional discovery in dynamic networks.

ACKNOWLEDGMENTS

This work was supported by grants NSF DMS-1228164 and NSA H98230-13-1-0241.

[1] S. E. Fienberg, J. Comput. Graph. Stat. **21**, 825 (2012).
[2] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, Found. Trends Mach. Learn. **2**, 1 (2009).
[3] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).
[4] B. Ball, B. Karrer, and M. E. J. Newman, Phys. Rev. E **84**, 036103 (2011).
[5] E. A. Leicht, G. Clarkson, K. Shedden, and M. E. Newman, Eur. Phys. J. B **59**, 75 (2007).
[6] P. Sarkar and A. Moore, in *Advances in Neural Information Processing Systems 18*, edited by Y. Weiss, B. Schölkopf, and J. Platt (MIT Press, Cambridge, 2006), pp. 1145–1152.
[7] P. D. Hoff, A. E. Raftery, and M. S. Handcock, J. Am. Stat. Assoc. **97**, 1090 (2002).
[8] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2007), pp. 687–696.
[9] M. Kolar, L. Song, A. Ahmed, and E. P. Xing, Ann. Appl. Stat. **4**, 94 (2010).
[10] E. Richard, P.-A. Savalle, and N. Vayatis, arXiv:1205.1406.
[11] S. Asur, S. Parthasarathy, and D. Ucar, ACM Trans. Knowl. Disc. Data **3**, 16 (2009).
[12] M. Raginsky, R. Willett, C. Horn, J. Silva, and R. Marcia, IEEE Trans. Inf. Theory **58**, 5544 (2012).
[13] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, Phys. Rev. E **83**, 066114 (2011).
[14] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, Data Min. Knowl. Disc. **22**, 493 (2011).
[15] S. Fortunato and M. Barthlemy, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).
[16] Y. Frishman and A. Tal, IEEE Trans. Vis. Comput. Graph. **14**, 727 (2008).
[17] D. Archambault, H. Purchase, and B. Pinaud, IEEE Trans. Vis. Comput. Graph. **17**, 539 (2011).
[18] S. Ghani, N. Elmqvist, and J.-S. Yi, Comput. Graph. Forum **31**, 1205 (2012).
[19] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner, Computer Graphics Forum **30**, 1719 (2011).
[20] J. S. Yi, N. Elmqvist, and S. Lee, Int. J. Hum.-Comput. Int. **26**, 1031 (2010).
[21] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction: With 200 Full-Color Illustrations* (Springer, New York, 2001), p. 533.
[22] Y. Koren, Comput. Math. Appl. **49**, 1867 (2005).
[23] U. Brandes, D. Fleischer, and T. Puppe, in *Graph Drawing*, edited by P. Healy and N. Nikolov, Lecture Notes in Computer Science Vol. 3843 (Springer, Berlin, 2006), pp. 25–36.
[24] K. Rohe and B. Yu, arXiv:1204.2296.
[25] K. Rohe, S. Chatterjee, and B. Yu, Ann. Stat. **39**, 1878 (2011).
[26] F. R. K. Chung, *Spectral Graph Theory* (American Mathematical Society, Providence, RI, 1997).

[27] D. D. Lee and H. S. Seung, Nature (London) **401**, 788 (1999).
[28] D. D. Lee and H. S. Seung, Adv. Neur. Inf. Process. Syst. **31**, 556 (2001).
[29] P. Paatero and U. Tapper, Environmetrics **5**, 111 (1994).
[30] K. Devarajan, PLoS Comput. Biol. **4**, e1000029 (2008).
[31] C. Ding, X. He, and H. D. Simon, in *Proceedings of the 2005 SIAM International Conference on Data Mining* (SIAM, Philadelphia, 2005), pp. 606–610.
[32] C. Ding, T. Li, and W. Peng, Comput. Stat. Data Anal. **52**, 3913 (2008).
[33] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, in *Proceedings of the 17th International Conference on World Wide Web* (ACM, New York, 2008), pp. 685–694.
[34] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, J. Comput. Appl. Math. **17**, 444 (2008).
[35] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, Comput. Stat. Data Anal. **52**, 155 (2007).
[36] Z. Chen and A. Cichocki, *RIKEN report,* 2005 (unpublished).
[37] P. O. Hoyer, in *Proceedings of the 2002 IEEE Workshop on Neural Networks for Signal Processing* (IEEE, Piscataway, NJ, 2002), pp. 557–565.
[38] P. O. Hoyer, J. Mach. Learn. Res. **5**, 1457 (2004).
[39] D. Cai, X. He, J. Han, and T. Huang, IEEE Trans. Pattern Analysis and Machine Intelligence **33**, 1548 (2011).
[40] H. Zou, T. Hastie, and R. Tibshirani, J. Comput. Appl. Math. **15**, 265 (2006).
[41] D. M. Witten, R. Tibshirani, and T. Hastie, Biostatistics **10**, 515 (2009).
[42] J. Guo, G. James, E. Levina, G. Michailidis, and J. Zhu, J. Comput. Appl. Math. **19**, 930 (2010).
[43] M. Chu, F. Diele, R. Plemmons, and S. Ragni, Technical Report (2004), http://users.wfu.edu/plemmons/papers/chu_ple.pdf.
[44] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004), pp. xiv, 716.
[45] C.-J. Lin, IEEE Trans. Neur. Netw. **18**, 1589 (2007).
[46] M. E. J. Newman, Phys. Rev. E **74**, 036104 (2006).
[47] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005).
[48] J. M. Kleinberg, J. ACM **46**, 604 (1999).
[49] *Matrix Analysis and Applied Linear Algebra*, edited by C. D. Meyer (SIAM, Philadelphia, 2000), Chap. 8.
[50] P. O. Perry and A. B. Owen, Ann. Appl. Stat. **3**, 564 (2009).
[51] G. G. Grinstein, C. Plaisant, S. J. Laskowski, T. O'Connell, J. Scholtz, and M. A. Whiting, *IEEE Symposium on Visual Analytics Science and Technology, VAST '08, 19–24 October 2008, Columbus, OH* (IEEE, Piscataway, NJ, 2008).
[52] A. A. Shaverdian, H. Zhou, G. Michailidis, and H. V. Jagadish, in *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration* (ACM, New York, 2009), pp. 74–82.
[53] Z. Shen and K.-L. Ma, in *Proceedings of the IEEE Pacific Visualization Symposium, 2008* (IEEE, Piscataway, NJ, 2008), pp. 175–182.
[54] Q. Ye, B. Wu, D. Hu, and B. Wang, in *Proceedings of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009* (IEEE, Piscataway, NJ, 2009), Vol. 2, pp. 413–417.
[55] M. Newman, A. Barabási, and D. Watts, *The Structure And Dynamics of Networks* (Princeton University Press, Princeton, 2006).
[56] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[57] G. Bounova and O. de Weck, Phys. Rev. E **85**, 016117 (2012).
[58] A. Clauset, C. Shalizi, and M. Newman, SIAM Rev. **51**, 661 (2009).
[59] J. Gehrke, P. Ginsparg, and J. M. Kleinberg, *SIGKDD Explorations* (ACM, New York, 2003), Vol. 5, pp. 149–151.
[60] J. Leskovec, J. Kleinberg, and C. Faloutsos, in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (ACM, New York, 2005), pp. 177–187.
[61] R. C. Feenstra, R. E. Lipsey, H. Deng, A. C. Ma, and H. Mo (unpublished).
[62] J. E. Stiglitz, W. B. Res. Obser. **11**, 151 (1996).
[63] R. R. Nelson and H. Pack (unpublished).
[64] G. Michailidis, *Handbook of Data Visualization* (Springer, Berlin, 2008), pp. 103–120.