Evolutionary Network Analysis: A Survey

CHARU AGGARWAL, IBM T. J. Watson Research Center
KARTHIK SUBBIAN, University of Minnesota

Evolutionary network analysis has found an increasing interest in the literature because of the importance of different kinds of dynamic social networks, email networks, biological networks, and social streams. When a network evolves, the results of data mining algorithms such as community detection need to be correspondingly updated. Furthermore, the specific kinds of changes to the structure of the network, such as the impact on community structure or the impact on network structural parameters, such as node degrees, also needs to be analyzed. Some dynamic networks have a much faster rate of edge arrival and are referred to as network streams or graph streams. The analysis of such networks is especially challenging, because it needs to be performed with an online approach, under the one-pass constraint of data streams. The incorporation of content can add further complexity to the evolution analysis process. This survey provides an overview of the vast literature on graph evolution analysis and the numerous applications that arise in different contexts.

Categories and Subject Descriptors: H.4 [**Information Systems Applications**]: Miscellaneous

General Terms: Algorithms

Additional Key Words and Phrases: Network analysis, temporal graphs, dynamic graphs

ACM Reference Format:
Charu Aggarwal and Karthik Subbian. 2014. Evolving network analysis: A survey. ACM Comput. Surv. 47, 1, Article 10 (April 2014), 36 pages.
DOI: http://dx.doi.org/10.1145/2601412

1. INTRODUCTION

Evolving networks arise in a wide variety of application domains, such as the Web, social networks, and communication networks. Networks are also sometimes referred to as *graphs* and will therefore be discussed interchangeably with graphs in this article. The recent interest in the area of dynamic social networks has led to a significant interest in the analysis of evolving networks [Aggarwal 2011]. Evolution analysis in graphs has applications to a number of different scenarios, such as trend analysis in social networks [Goetz et al. 2009; Leskovec et al. 2007; Wang and Chen 2009; Yan et al. 2012; Aggarwal and Subbian 2012], and dynamic link prediction [Acar et al. 2009; Tylenda et al. 2009; Sarkar et al. 2012; Sarukkai 2000]. Most real-life networks evolve in a wide variety of ways that lead to different kinds of evolution semantics.

Evolving network analysis can be generally divided into one of two distinct categories. These categories, although distinct, do overlap with one another from a methodological

This work was supported by the Army Research Laboratory, under Cooperative Agreement Number W911NF-09-2-0053.
Author's addresses: C. Aggarwal, IBM T. J. Watson Research Center, 1101 Kitchawan Rd, Yorktown Heights, NY, 10598; email: charu@us.ibm.com; K. Subbian, Computer Science Department, 200 Union St SE, Minneapolis, MN, 55455; email: karthik@cs.umn.edu.

DOI: http://dx.doi.org/10.1145/2601412

perspective, especially in the context of a few key problems such as community detection, which can be seen as a "bridge" between these two modes of analysis:

—*Maintenance Methods:* In these cases, it is desirable to maintain the results of the data mining process continuously over time. For example, the results of a classification and clustering method will evolve as the structure of the graph changes over time. Therefore, the results of the methods will become stale over time, and the goal is to maintain the freshness of the end results. Correspondingly, it is desirable to provide methods that can maintain these results continuously and incrementally over time.
—*Analytical Evolution Analysis:* In these cases, it is desirable to directly *quantify* and *understand* the changes that have occurred in the underlying network. The main point to remember is that such models are focused on modeling the change, rather than correcting or adjusting for the staleness in the results of data mining algorithms on networks. Direct evolution analysis is closely related to the problem of outlier detection in temporal networks because temporal outliers are often defined as (abrupt) change points.

It should also be pointed out that the community detection problem is special because it falls into both categories. This is because a clustering can often be viewed as an unsupervised model of the entire network, especially when it is used in the context of a generative methodology. Therefore, the temporal variation of the generative behavior of the network, often provides unique insights into the overall network evolution. As shown in Gupta et al. [2011b], a tightly integrated generative framework can be used to model the maintenance of evolving clusters and also perform the evolution analysis.

Not all networks evolve equally fast or have links that are added at the same rate. For example, in email networks, transient links are added to the network on the time scale of seconds (corresponding to emails between participant nodes), whereas in bibliographic networks, edges are added to the network on the time scale of weeks or months. Correspondingly, these scenarios require different kinds of analysis:

—*Slowly Evolving Networks:* In these cases, the network evolves slowly over time, and *snapshot* analysis can be used very effectively. In these situations, snapshots of the network at two distinct times t_1 and t_2 are used for analysis, and therefore offline analysis can be performed directly.
—*Streaming Networks:* Many networks that are created by transient interactions, such as email or telecommunication networks, can be represented as graph streams. Graph streams typically require *real-time* analytical methods. This scenario is far more challenging because of the computational requirements and the inability to hold the entire graph on the disk. Such scenarios could arise in the context of streams of objects [Aggarwal et al. 2010], edges [Zhao et al. 2011], or linked data streams [Le-Phuoc et al. 2012].

The categorizations of the different scenarios for network evolution analysis may be present in any arbitrary combination. For example, model maintenance methods can be studied both in the snapshot and the streaming scenario, where the latter is significantly more difficult than the former. Similarly, evolution analysis methods can also be studied in both contexts. In some cases, content associated with nodes and links can be used to further enhance evolutionary analysis. This survey provides an overview of the wide gamut of methods that can leverage the richness of the different scenarios in the network analysis domain.

1.1. Semantics of Network Evolution in Different Domains

Network evolution has different kinds of semantics in different application domains. In the following, a brief discussion of these varying semantics is provided. The application Section 5 contains more details of how evolution analysis can be leveraged for application-specific insights.

- *Web Semantics:* The Web continually evolves over time as new Web pages and links are created, and old ones are deleted. This leads to numerous applications of network evolution analysis such as visualization of Web ecologies, discovery of inconsistencies in crawling, and external events based on user click stream behavior [Chi et al. 1998; Chan et al. 2008; Dorogovtsev and Mendes 2003; Papadimitriou et al. 2010].
- *Social Network Semantics:* New links and nodes are continuously created in a wide variety of formal and informal social networks as new actors join the social network, and new friendships are created. Because key changes in the network are often caused by external events, this leads to a number of important applications such as event and anomaly detection [Doreian and Stokman 2013; Kumar et al. 2006; Aggarwal and Subbian 2012; Silva and Willett 2008; Tong et al. 2008c].
- *Biological Network Semantics:* Biological networks are typically expressed in the form of interaction or correlation networks. For example, in a protein-protein interaction network, a node corresponds to a protein, and an edge corresponds to an interaction between the two proteins [Vázquez et al. 2002]. Many biological functions of organisms are dependent on interactions between two proteins. Interestingly, the structure of this network, and the interactions, changes with the age of the protein. This has a direct impact on biological functioning. Providing an understanding of the nature of the evolution therefore provides the key to numerous insights. In gene-expression networks [Stuart et al. 2003], the similarity in interactions (edges) of different genes (nodes) evolves over time in response to external factors such as clinical drugs in the context of oncology.
- *Metabolic Network Semantics:* In metabolic networks, the nodes correspond to different intermediate products and enzymes in animal metabolism, and the edges correspond to the transformations between them. A disruption in the natural metabolic network typically has a direct impact on the organism itself. A classical example of this is the development of type-2 diabetes, in which the insulin-based metabolism pathways are disrupted gradually, with the development of insulin resistance [Beyer et al. 2010]. An insight into the evolution of such networks in an individual leads to a better understanding of the development of different kinds of diseases.

It should be pointed out that many of the same evolution analysis methods can be applied to scenarios as diverse as social and biological networks [Asur et al. 2007]. The application section also provides further discussion of the semantics of evolution analysis in the context of different domains.

1.2. Related Surveys and Differences

One of the earliest surveys on network evolution analysis may be found in Bilgin and Yener [2006], although this work is quite outdated at this point because the majority of the network evolution analysis research has been performed in the last decade, with the greater popularization of online social and biological networks. A more recent work is found in Spiliopoulou [2011]. This survey provides a good overview of community evolution in social networks, along with a brief discussion of evolution laws, although the subject of network evolution is much broader. Most data mining problems such as clustering, classification, and outlier detection can be generalized to network data, with additional problem definitions in fields such as link prediction and influence

analysis. This survey provides an integrated treatment of evolution analysis in the context of these different topics, along with a discussion of streaming analysis and different applications. This survey also provides pointers to Spiliopoulou [2011] where more details on topics such as community evolution are discussed.

A survey on managing and mining streaming graphs is found in Zhang [2010]. The survey in Zhang [2010] primarily focuses on theoretical methods for counting triangles, graph matching, and graph distances in the streaming model. Readers are advised to refer to this survey for discussions on this topics. This survey focuses on traditional data mining problems such as clustering, dense pattern mining, and classification in the context of evolutionary graphs. The material discussed in this survey is complementary to the streaming survey proposed by Zhang [2010].

Finally, a number of surveys on static network analysis are available in recent books [Aggarwal 2011; Aggarwal and Wang 2010]. These include surveys on topics such as link prediction [Hasan and Zaki 2011], influence analysis [Sun and Tang 2011], graph classification [Bhagat et al. 2011; Tsuda and Saigo 2010], and statistical properties of real-world graphs, [Chakrabarti et al. 2010; McGlohon et al. 2011]. These surveys are mostly focused on static networks, and our survey builds on the static model formulations discussed in these surveys. For example, it is discussed how communities evolve in the context of community detection, links evolve in the context of link prediction, and node labels evolve in the context of graph classification. The laws of evolution are discussed in Chakrabarti et al. [2010] and McGlohon et al. [2011]. Our survey discusses how many of the laws of evolution can be derived from (and related to) the scale-free model in a systematic way, which is not addressed in these surveys. While many of these surveys describe many different aspects of evolution analysis, their general focus is different, with the exception of the survey of Spiliopoulou [2011]. Our survey provides a broader treatment of the subject, along with many different data mining problems, streaming scenarios, and application domains. We also address the problem of community detection, which is the main focus of Spiliopoulou [2011]. We also discuss several evolution methods for community detection, not discussed in Spiliopoulou [2011], along with pointers to some of the aspects discussed in Spiliopoulou [2011]. Many related aspects of graph summarization will also be discussed, not covered in Spiliopoulou [2011].

1.3. Survey Organization

This survey is organized as follows. Methods for maintaining time-evolving models are discussed in Section 2. Methods for change analysis in evolving graphs are discussed in Section 3. The evolutionary clustering methods are described in both Sections 2 and 3, depending upon whether the focus of the clustering is maintenance or analytical. Nevertheless, a significant cross-usability exists in the clustering methods described in the two sections. Furthermore, each of these sections contains subsections on both the snapshot-based algorithms and streaming methods. The use of content for enhancing evolution analysis is discussed in Section 4. Numerous applications of evolution analysis are discussed in Section 5. The conclusions and summary are discussed in Section 6.

2. MAINTAINING TIME-EVOLVING MODELS

Numerous time-evolving models exist for different kinds of graph analysis problems such as clustering, classification, influence analysis, and link prediction. This section provides an overview of the different models that are used for these problems. The description of this section is divided into slowly evolving and streaming networks.

2.1. Slowly Evolving Networks

In many networks such as bibliographic or other Web-based networks, significant changes occur in the network on the time-scale of a few days or months. In these cases, snapshot-based methods are used for the analytical process. Given networks at times t_1 and t_2, the results of the data mining algorithm are adjusted at each snapshot by incrementally adjusting the results from the previous snapshot.

2.1.1. Clustering and Community Detection. One of the earliest methods for evolutionary clustering was proposed in Chakrabarti et al. [2006]. The *evolving clustering* method proposed in Chakrabarti et al. [2006] balances two important objectives while performing the online clustering process: (1) the newly formed data clusters should accurately reflect the data at the current time step, and (2) the clusters formed at current time step should be closely similar to the clusters formed at previous time step. The first criteria is referred to as *consistency,* whereas the second is referred to as *smoothness*. Hence, the clustering algorithm performs a trade-off between the cost of maintaining the clusters accurately at the current time step at the cost of deviating from the historical data.

The evolutionary spectral clustering approach proposed by Chi et al. [2009] uses a cost function that includes both consistency and smoothness terms. The consistency term in spectral clustering is maximizing $Tr(X^T WX)$ with respect to the graph embedding variables X and graph similarity matrix W. The smoothness objective expects a certain level of temporal smoothness between X_{t-1} and X_t. The smoothness can preserve either the Cluster Quality (PCQ) or Cluster Membership (PCM). Let Z_t be an association matrix of size $n \times k$ where n is the number of data points and k is the number of clusters. Entries of Z_t are either zero or one, denoting the membership of the data points to the cluster. Let $\widetilde{Z}_t$ be normalized matrix $\mathbf{Z}_t$ by the cluster size. In addition, as the data points are partitioned into clusters, $\mathbf{Z}_t$ turns out to be orthonormal, i.e., $\langle \widetilde{Z}_t, \widetilde{Z}_t \rangle = I_k$. The temporal smoothness objective in PCM is to maximize the spectral cut objective at time $t-1$, while the same association matrix $\widetilde{\mathbf{Z}}_t$ retains reasonable clustering quality at time $t-1$. More formally, $Tr(\widetilde{Z}_t^T W_{t-1} \widetilde{Z}_t)$ denotes the temporal smoothness objective where cluster quality at time $t-1$ is measured w.r.t current associations $\widetilde{Z}_t$. In PCM, the temporal smoothness is measured in terms of maintaining the cluster memberships rather than cluster quality. The cluster memberships do not change if the association at time $t-1$ is close enough to t in terms of a distance measure. An appropriate distance measure must satisfy the rotation invariance property because the spectral cut objectives are rotation invariant. A distance measure that satisfies this condition is the norm difference between the projection matrices. Formally, the distance between X_t and X_{t-1} is then given by $\frac{1}{2}\|X_t X_t^T - X_{t-1} X_{t-1}^T\|^2$, where X_t is a relaxed version of Z_t.

A forgetting factor is incorporated in Xu et al. [2010], in order to allow the approach to adjust better for the evolution in the underlying network. This kind of decay factor is quite common in all forms of dynamic evolutionary analysis in different domains. It has been discussed in Ning et al. [2007] how such models may be used to perform evolutionary analysis of blogs. Blogs are a particularly suitable domain for this kind of analysis because of the continuous updates to the structure of the graph and the relatively fast evolution that may occur in response to a news event of interest.

The temporal smoothness principle is used in conjunction with a particle-and-density approach [Kim and Han 2009] for creating clusters from time-evolving graphs. This approach uses temporal smoothing as a mechanism to detect evolving community structures in co-authorship and sports networks. Their formulation has both snapshot and temporal smoothing quality as part of the objective. The snapshot quality is simply

Table I. Evolutionary Clustering Methods

Class of Methods	Related Work
Spectral	[Chakrabarti et al. 2006] [Chi et al. 2009] [Ning et al. 2007] [Tang et al. 2008]
Probabilistic	[Xu et al. 2012] [Gupta et al. 2011b] [Lin et al. 2008] [Sun et al. 2010]
Density-based	[Falkowski et al. 2008] [Kim and Han 2009]
Matrix Factorization	[Wang et al. 2012] [Sun et al. 2006]
Modularity	[Takaffoli et al. 2013], [Görke et al. 2010]
Information Theoretic	[Sun et al. 2007] [Ferlez et al. 2008]
Pattern Mining	[Ahmed and Karypis 2012] [Berlingerio et al. 2009a]
Others	[Bogdanov et al. 2011] [Palla et al. 2007]

the density-based clustering quality using the edge similarity between the vertices. The temporal smoothing quality is the one-dimensional Euclidean distance between a pair of vertices at current and previous time instants. This approach does not require any predefined number of clusters and can dynamically detect and adapt to any number of clusters based on the current and previous time points. An important aspect of this approach is that it detects evolving communities, which is a collection of clusters that exist across a set of snapshot graphs for a defined period. The clusters at each snapshot are called a nano-community, and an approximate l-clique-by-clique approach combines several of these nano-communities to form an evolving community structure. Another density-based approach for incremental community detection, with the use of principles from multidimensional density-based clustering methods such as DBSCAN, is the *DENGRAPH* algorithm [Falkowski et al. 2008].

In many cases, the underlying networks are heterogeneous, in which the links and nodes may be of different types. A method known as *ENetClus* [Gupta et al. 2011b] was proposed, in which a probabilistic mixture model was used to characterize the underlying clusters. This probabilistic mixture model is used in conjunction with a smoothness criterion to determine soft clusters over successive snapshots. Different properties of the clusters, such as consistency and clustering quality, are also explored in Gupta et al. [2011b] to characterize the nature of the evolution of the clusters. This is actually a tightly integrated model that can simultaneously perform clustering and evolution analysis. The popular evolutionary clustering methods are provided in Table I. A discussion of evolution analysis methods for community detection may be found in Spiliopoulou [2011].

2.1.2. Low Rank Approximation. Low rank approximation is vital for identifying the community structures and anomalies in networks by applying these methods on the adjacency matrix [Ning et al. 2007; Sun et al. 2006]. Many methods have been proposed for low rank approximation such as Singular Value Decomposition (SVD), matrix factorization, CUR, Compact Matrix Decomposition (CMD), and more recently *Colibri* [Drineas et al. 2006; Jolliffe 2005; Seung and Lee 2001; Sarwar et al. 2002; Sun et al. 2007; Tong et al. 2008a]. Each of these different kinds of low rank approximations has been shown to be updatable in the dynamic scenario.

For the problem of SVD, it was shown in Sarwar et al. [2002] how dynamic updates may be designed. Although the base matrix used was a user-item matrix in the context of a recommendation application, this can be viewed as an evolving user-item graph, and the approach can also be applied to the case of adjacency matrices of generic graphs. SVD can be easily extended to the dynamic scenario by observing that the iterative algorithm is very fast when the starting point is already a good approximation of the optimal solution. This general principle is also true of matrix factorization [Seung and

Lee 2001]. Here, the $n \times n$ adjacency matrix is factorized as $A \approx UV$. In this case, U and V are low-rank $n \times k$ and $k \times n$ matrices. The solution to this problem is a set of iterative updates of the following form, for appropriately defined functions f and g:

$$U^{t+1} = f(U^t, V^t), \quad V^{t+1} = g(U^t, V^t).$$

Therefore, when an approximate solution is already available for U and V, a small number of iterations provides an optimal solution.

A similar approach has also been shown to work for the *Colibri* method [Tong et al. 2008a], in which the decomposition is $A \approx LMR$. Here, L is a $n \times k$ matrix, for which each of the k columns is a subset of the columns of A. M is a $k \times k$ matrix, and R is a $k \times n$ matrix. In this case, the matrix M from the previous time point is used to estimate the L matrix at current time. After estimating L, the matrix M is updated, and R is back computed. The dynamic *Colibri* method is five times faster than its static counterpart and two orders of magnitude faster than most of the other methods in the dynamic case.

2.1.3. Classification. In the problem of node classification, the labels of a subset of the nodes in an evolving network are available and are used to dynamically predict the labels of the remaining nodes. The work in Aggarwal and Li [2011] proposes a dynamic method for classification of content-based networks. In this technique a random-walk based approach is used in which the fraction of nodes visited belonging to each class is used to determine the class label. The random walk is performed on a graph that effectively combines the structure and content for classification. To do so, the network is augmented with additional nodes and edges. Specifically, a pseudo-graph is created that has one node corresponding to each node in the original network and also has one node for each keyword in the network. A structural node is connected to a keyword node if the keyword is contained in that node. The links between structural nodes are maintained in the same way as the original network. By varying the weights of the structural and content-based links, it is possible to determine the relative importance of content and structure in the random walk. A dynamic inverted index is maintained to efficiently perform the random walk. The majority label among nodes visited during the classification is reported as the relevant one. The same basic problem can also be studied in the entity-relationship graph setting [Güneş et al. 2013], in which nodes correspond to entities and edges correspond to relationships. This approach uses genetic algorithms as an optimization approach. The assignment of node labels to nodes is determined by optimizing the error of the classification using genetic algorithms.

2.1.4. Link Prediction. Link prediction is one of the most fundamental problems in the analysis of networks because it directly predicts links in the future based on previous trends [Liben-Nowell and Kleinberg 2007; Sarukkai 2000; Taskar et al. 2003; Al Hasan et al. 2006; Popescul and Ungar 2003]. The Web and social networks are continuously evolving over time, with new nodes and links being added over time. While links are also deleted at times, the addition of links is a more common occurrence in social networks. Therefore, the link prediction problem attempts to determine the most likely links that will be added to the network in the future. It is often not studied directly in the context of continuous or incremental scenarios because it is assumed that we have a given network at a specific moment in time, and we are trying to predict most likely links to appear *at any point* in the future from this *single snapshot* of the network. However, some of the recent work incorporates the temporal component more directly by using multiple snapshots and designs either (continuously) dynamic methods for link prediction [Aggarwal et al. 2012b; Sarkar et al. 2012; Huang and Lin 2009; Tylenda et al. 2009; Kolar et al. 2010] or tries to determine the time at which a link will appear

in the future [Sun et al. 2012]. It has been shown convincingly in Huang and Lin [2009] and Tylenda et al. [2009] that the incorporation of the continuously evolving behavior of the network clearly improves the behavior of link prediction problems. A nonparametric link prediction algorithm for a sequence of graph snapshots over time is proposed in Sarkar et al. [2012]. The model predicts links based on the features of its endpoints, as well as on those of the local neighborhood around the endpoints. This algorithm can adjust for different types of temporal dynamics, such as growing or shrinking communities in the network. In cases where the snapshots of the network are not available, but only attributes of the network are available at different snapshots, it has been shown how to estimate the network links at different moments in time [Kolar et al. 2010]. Link prediction methods are extremely useful for recommendation and collaborative filtering problems [Backstrom and Leskovec 2011; Huang et al. 2005]. This is discussed in some detail in the applications section. Because this article is not specifically focused on link prediction (which is a very broad topic in its own right), we omit a detailed discussion of the methods and refer the reader to Hasan and Zaki [2011] for a detailed survey.

2.1.5. Tensor Factorization. Tensors are higher order extensions of matrices, data cubes, or multidimensional arrays. Typically, a time evolving graph is modeled as a third-order tensor [Acar et al. 2009; Dunlavy et al. 2011; Sun et al. 2006] where the first two dimensions represent an adjacency matrix and the third dimension captures the sequence of such adjacency matrices representing the evolution of the network. Tensor factorization has been used in link prediction problems where the evolution of the network is given until time t, and the links that will be formed at time $t+1$ have to be predicted [Acar et al. 2009]. The simplest approach is to collapse the time-dimension of the tensor into a decay-weighted second-order matrix. Standard low rank approximation techniques such as SVD can be used to predict the future links. Unfortunately, such an approach leads to significant loss of information because it does not explicitly account for the time dimension in the modeling process.

A different approach is to factorize a tensor using tensor decomposition techniques, such as *Candecomp/Parafac (CP)* [He et al. 2005]. A three-way tensor $\mathcal{Z}$ of size $M \times N \times T$, and its K-component decomposition is given as, $\mathcal{Z} \approx \sum_{k=1}^{K} \lambda_k \mathbf{a}_k \circ \mathbf{b}_k \circ \mathbf{c}_k$. Here, the factors are $\mathbf{a}_k \in \Re^M$, $\mathbf{b}_k \in \Re^N$, $\mathbf{c}_k \in \Re^T$; $\circ$ denotes an outer product and λ_k denotes the scalar weight of the k-th component. Unlike SVD, factors are nonorthogonal to each other but shown to be unique within a permutation and scaling [Indyk et al. 2000]. If the tensor is sparse, then the complexity of these methods are in the order of the number of entries in the tensor. The main advantage of the factorization is that once the evolving graph is described in terms of a smaller number of variables, conventional tools such as temporal regression analysis can be used efficiently on this smaller number of variables.

To determine the approximate matrix factorization, the least square error of approximating the matrix entries is optimized. Most of the techniques use alternating least squares to compute the tensor factors. The $\mathbf{c}_k$ factor determines the temporal profile of other two factors $\mathbf{a}_k$ and $\mathbf{b}_k$ of $\mathcal{Z}$. One can compute the likelihood of i linking to j via a matrix computed as:

$$S = \sum_{k=1}^{K} \gamma_k \lambda_k \mathbf{a}_k \mathbf{b}_k^T.$$

Here, γ_k is a simple linear scaling function with value $\sum_{t=l}^{T} c_k(t)$. The parameter l denotes the number of previous time instances to be considered. The main disadvantage of this method is that it is not incremental and is therefore expensive to maintain.

The *Dynamic Tensor Analysis (DTA)* proposed in Sun et al. [2006] does not require all the snapshots to be available upfront for the analysis. The idea behind this approach is that the covariance matrix can be computed incrementally, and factors can be computed directly from the covariances relatively quickly without storing the historical snapshots of networks. Also, this model incorporates a forgetting factor to specify the relative importance of the covariances of the historical snapshot compared to the current snapshot covariance. The *DTA* method is efficient compared to offline tensor analysis techniques in both space and time. However, in streaming scenarios, the edges may arrive very fast. On the other hand, the change in the covariance matrix is quite small, and the expensive process of diagonalization for every new tensor can be expensive and not necessary. A *Streaming Tensor Analysis (STA)* [Sun et al. 2008, 2006] approach is proposed to do approximate and incremental tensor factorizations. The main idea here is to use an online PCA-like technique, where each row of an incoming tensor is used to approximate the factors using the reconstruction error. It is only when the errors are sufficiently large that the factors are updated.

2.2. Streaming Scenario

A particularly challenging scenario is the case of streaming graphs, when a large number of edges representing interactions are continuously received over time and that are superposed over a much larger network. An example of such a scenario would be a Twitter post stream, in which many posts are continuously received over time. Because the streaming model is new, no streaming methods exist for many of the techniques discussed in the previous section. Therefore, this is a fertile area for future research.

2.2.1. Clustering and Dense Pattern Mining. A method has been proposed in Aggarwal et al. [2010] in which small graphs (or edges) are clustered in streaming fashion with the use of a partitioning approach. A sketch structure is used to maintain the large number of distinct edges in the graph stream in a memory efficient way, although at some potential loss of accuracy. Theoretical bounds are proposed in Aggarwal et al. [2010] on the loss of accuracy resulting from such an approach. This method has also been extended to scenarios where side information such as content is associated with the incoming objects [Zhao and Yu 2013]. A second method given in Aggarwal et al. [2011] proposes a reservoir sampling method for clustering graph streams, and it has also been shown how the method can be used for temporal outlier detection. This method is discussed in some detail in Section 3.2. Subsequently, a number of enhancements over this basic method were proposed in Eldawy et al. [2012]. This approach allows for both edge additions and deletions. Incorporating the ability to delete edges is important in streaming scenarios when the clustering is performed over a sliding window of edges, and therefore edges are deleted from the tail end of the sliding window. A method for dynamic community discovery in graph streams was proposed in Lai et al. [2013]. Other methods for near linear time community detection in graphs are proposed in Leung et al. [2009] and Raghavan et al. [2007], although these methods are not specifically focused either on evolutionary analysis or on graph streams. Distributed methods for streaming graph partitioning are presented in Stanton and Kliot [2012].

The problem of clustering is closely related to dense pattern mining. This is because clusters are dense patterns in the data [Aggarwal et al. 2010]. For example, in a graph stream, which is composed of objects derived from a bibliographic network, co-occurring nodes correspond to authors who often write publications together. A dense group of nodes is defined as ones that co-occur together frequently, and the density of the edges between this group of nodes is high. A min-hash approach is used to determine the relevant groups of nodes in an online fashion.

2.2.2. Classification. A method for classification of graph streams was proposed in Aggarwal [2011]. This method attaches a label with each small graph, which is superposed on a potentially large graph. It is assumed that the edges in the graph stream are received in *arbitrary* order, so that the edges for a particular small graph may not be received continuously. A min-hash model is used to determine structural patterns that are related to the labels. For a given graph, it is determined which structural patterns are most relevant. These are then used for the purposes of classification.

Recently, the problem of streaming classification has also been extended to different settings such as imbalanced data distributions [Pan and Zhu 2013] and semisupervised learning [Pan et al. 2013]. In particular, the concept of hashing [Li et al. 2012; Guo et al. 2013] can be used to create a compressed representation for streaming graph classification when combined with kernel methods. The idea is to create subtree hash kernels in real time, which are then leveraged for more effective classification. The work in Aggarwal [2011] also uses hashing, but in the context of a sketch-based approach, to explicitly model dependencies between sketched subgraphs and different classes.

2.2.3. Miscellaneous Problems. An important class of algorithms in this context is page rank analysis, in which it is desirable to estimate the page rank on a dynamic evolving graph stream [Das Sarma et al. 2008; Bahmani et al. 2010; Desikan et al. 2005]. The method in Das Sarma et al. [2008] is able to estimate the page rank distribution, the mixing time, and the conductance of the graph. The method in Bahmani et al. [2010] designs a method for real-time estimation of the personalized page rank in graph streams.

The problem of model maintenance has been studied in the context of query processing [Zhao et al. 2011]. For the query processing problem, it has been shown how a partitioned sketch model can be used to respond to edge frequency queries. Another recent work shows how to perform continuous subgraph queries over data streams [Choudhury et al. 2011; Wang and Chen 2009]. The problem of graph matching in streams has been addressed in McGregor [2005]. For a given graph, the *Maximum Cardinality Matching (MCM)* problem is to find the largest set of edges such that no two adjacent edges are selected. More generally, for an edge-weighted graph, the Maximum Weighted Matching (MWM) problem is to find the set of edges whose total weight is maximized subject to the condition that no two adjacent edges are selected. Although these problems are well studied in the static scenario, they can be prohibitively expensive if random access to the data is not assumed. Clearly, this is not possible in the streaming scenario. The work in McGregor [2005] proposes a $1/(1+\epsilon)$ approximation algorithm for the maximum cardinality version and a $1/(2+\epsilon)$ algorithm for the maximum weighted matching problem. The algorithm requires $O(|V|)$ space and a constant number of passes in which the edges are streamed in arbitrary order. Thus, this is a *weakly* streaming model in which a constant number of passes is assumed rather than a single pass.

3. ANALYTICAL EVOLUTIONARY ANALYSIS

Graphs evolve over time as new edges are added and old ones are deleted. It is important to provide different kinds of insights about the nature of the underlying evolution. This is different from *maintenance models* because the focus is not about replenishing the staleness of the model, but instead understanding the overall dynamics of the entire evolution of the graph. Nevertheless, there are clear connections between the two classes of problems because many maintenance models for problems such as evolutionary clustering are used to understand the nature of the underlying evolution.

Because edge addition is more common in many scenarios such as the Web and social networks, the typical trend in graphs is *densification with shrinking diameters*

[Leskovec et al. 2005b]. Change analysis can be characterized with a variety of different measures such as centrality, community behavior, minimum description length (MDL), shortest paths, or rules. In this section, we provide an overview of the wide variety of methods with which network evolution can be understood effectively. Many measures of the network, such as centrality, can be determined by using the methods discussed in Tong et al. [2008b]. The associated computational challenge is a major difference in the two network settings of fast and slowly evolving networks.

3.1. Slowly Evolving Networks

These cases are either based on snapshot analysis or are based on study of real-world networks in which the analysis is incremental but still relatively slow over time.

3.1.1. Large Scale Models and Laws for Network Evolution. Numerous large-scale models have been proposed for understanding "typical" evolutionary behavior of social networks. These methods focus on the key laws that are generally true across a wide variety of social networks, rather than on methods for analyzing a specific social network scenario. Many of the laws of evolution in social networks are discussed in Chakrabarti et al. [2010], McGlohon et al. [2011], and Albert and Barabási [2002]. In this survey, we provide a coherent presentation of how many of these laws relate to one another systematically, starting with the basic notion of preferential attachment. Most evolution analysis laws are derived analytically from this basic notion:

(1) *Preferential Attachment:* The likelihood of receiving new edges increases with the node's degree. If $\pi(k)$ is the probability that a node attaches itself to a node i with degree k, then the probability $\pi(k)$ is related to k as follows:

$$\pi(k) \propto k^{\alpha}. \tag{1}$$

In some models, a constant A is added to the right-hand side of Equation (1) to account for the fact that isolated nodes may also receive edges. Here, α is a parameter whose value is dependent on the underlying domain. In some domains, such as citation networks, a scale-free assumption is used in which $\alpha \approx 1$, and therefore the proportionality is linear.

(2) *Effect of Nonlinear Preferential Attachment:* The number of nodes $N_k(t)$ with $(k-1)$ incoming edges in a directed network at time t can be quantified with the use of a rate equation approach [Krapivsky et al. 2000], by leveraging the preferential attachment rule:

$$\frac{\mathrm{d}N_k(t)}{\mathrm{d}t} = \frac{1}{M_{\alpha}(t)} \cdot [(k-1)^{\alpha} \cdot N_{k-1}(t) - k^{\alpha} \cdot N_k(t)] + \delta_k. \tag{2}$$

The first term corresponds to new nodes that connect to other nodes with $(k-1)$ edges (and thereby increasing the value of $N_k(t)$). The second term corresponds to nodes for which the degree increases from k to $k+1$, which decreases the value of $N_k(t)$ (and therefore the term is negative). The third term corresponds to the addition of new nodes with a single outgoing edge. The term $M_{\alpha}(t) = \sum_k^{\alpha} N_k(t)$ is the αth moment of $N_k(t)$. In the case of linear preferential attachment with $\alpha = 1$, it can be shown that the degree distribution of the nodes is given by:

$$P(k) \propto k^{-\gamma}. \tag{3}$$

Here $P(k)$ is the number of nodes with degree k, and γ is a parameter determined by the rate equation. For the case when α is either less or greater than 1, the power law degree distribution continues to be true, although in somewhat different form. However, the scale-free nature of the network is destroyed by the nonlinear preferential attachment.

(3) The scale-free model only incorporates addition of edges by adding new nodes to the system. In real systems, other events such as addition, rewiring of edges, and the removal of nodes and edges can impact the network. In cases where internal edges are added or rewiring occurs, the power law has been shown [Albert and Barabási 2000] to follow a generalized degree distribution:

$$P(k) \propto (k + \kappa(p, q, m))^{-\gamma(p,q,m)}. \tag{4}$$

Here, $\kappa(p, q, m)$ and $\gamma(p, q, m)$ are functions of the probability p that m edges are added with a specific distribution for probabilities for the end points and the probability q that m edges are rewired. A model has also been proposed in Dorogovtsev et al. [2000] that addresses the case when new internal edges are added and old edges are removed.

(4) *Competition in Evolving Networks:* In the scale-free model, a natural outcome is that the oldest nodes always have the highest number of edges. However, this is generally not true in many real networks, such as the Web, where many nodes can pick up a large number of edges in a small amount of time. It has been argued in Bianconi and Barabási [2001] that real networks have a competitive aspect in which some nodes draw away edges from others. Therefore, a generalized preferential attachment model [Bianconi and Barabási 2001] has been proposed in which young nodes with a few edges can acquire many edges at a high rate on the basis of a fitness parameter. This fitness parameter quantifies the ability of nodes to compete for edges from other nodes.

(5) *Copying Mechanisms:* To explain the power law behavior of the World Wide Web, models have been proposed in Kleinberg et al. [1999] and Kumar et al. [2000], according to which new pages on a specific topic copy links from existing pages on the same topic. For a new node, a "prototype" node is picked randomly. The destination of the ith edge of the new node is either chosen randomly with probability p, or it is chosen as the ith destination node of the prototype node with probability $(1 - p)$. Note that the second part of this process increases the probability of high-degree nodes receiving new edges. Thus, this mechanism provides an intuitive explanation for power law degree distributions of the Web.

The work of Leskovec et al. [2005b] studies the laws of evolving networks in a variety of real-world networks. A number of different citation and affiliation graphs (derived from broader bibliographic networks such as *Arxiv* or the *KDD CUP 2003* datasets [Gehrke et al. 2003]) were used. The main observations are as follows:

(1) The graphs gradually densify over time, with the number of edges growing superlinearly with the number of nodes. If $n(t)$ is the number of nodes in the network at time t and $e(t)$ is the number of edges, then the network exhibits the following *densification power law*:

$$e(t) \propto n(t)^{\alpha}. \tag{5}$$

Here, α is an exponent that lies strictly between 1 and 2. The value of $\alpha = 1$ corresponds to a network where the node degree does not change, whereas the value of $\alpha = 2$ corresponds to a network in which the degree is a constant fraction of the total number of nodes.

(2) As the network densifies, the average distances between the nodes shrink over time. The effective diameter of a graph over time is necessarily bounded from below, and the decreasing patterns of the effective diameter in the experimental studies were consistent with convergence to some asymptotic value.

(3) As the network densifies over time, a giant connected component emerges. In all the studied networks, most of the nodes belonged to the giant connected component

after a few years. It was also observed that the diameter shrinking phenomenon was not dependent on this fact because the shrinking of the network continued even after maturation of the network, in which most of the nodes belonged to one component. The emergence of a giant connected component is consistent with the principle of preferential attachment, in which newly incoming edges are more likely to attach themselves to the largest component in the network.

The work in Leskovec et al. [2005b] also proposes a data generator that is dependent on and consistent with these properties. This model, referred to as the *Forest Fire Model*, is based on having new nodes attach to the network by "burning" through existing edges in epidemic fashion.

A large-scale experimental study of microscopic evolution of social networks is provided in Leskovec et al. [2008]. Four large online social networks *Flickr*, *delicious*, *Yahoo! Answers,* and *LinkedIn* were analyzed with full temporal information about node and edge arrivals in Leskovec et al. [2008]. This was the first large-scale experimental study of the preferential attachment principle in real networks, given the unavailability of sufficient data prior to this point. It was shown that there were minor differences in the exponent α of the preferential attachment rule $\pi(k) = k^{\alpha}$ among the four networks and also between low-degree and high-degree nodes within a network. In all cases, the exponent was close to 1, which means that the attachment could be treated as essentially linear. The work in Leskovec et al. [2008] studies the fraction of edges initiated by nodes of a certain age. It was shown that there is a spike at age 0, when people join the network to create an edge. Subsequently, the level of activity is relatively uniform over time. A maximum likelihood model was constructed in which the combined effect of the edge and the degree distribution was studied, along with models that study purely the effect of age a or purely the effect of degree k:

$$\pi(k) \propto k^{\alpha} \cdot a^{\beta}. \tag{6}$$

The first term in the product on the right-hand side corresponds to the node degree (preferential attachment), whereas the second term corresponds to the age of the node, with β as the exponent of that term. Different models can be constructed by predeciding *some* of the values of α and β to specific values (including 0) and learning the others in a data-driven manner using maximum likelihood estimation. Four different models were studied, corresponding to (i) proportionality to k^{α}, (ii) proportionality to k with a certain probability and randomly picked otherwise, (iii) proportionality to a^{β}, and (iv) proportionality to $k \cdot a^{\beta}$. It was shown that the last model, which uses linear degree-based attachment ($\alpha = 1$) and some impact of age,[1] typically performed the best in most cases. One observation is that attachment often has a nonlocal component to it, since two nodes that share many friends in common are more likely to form a link between them. Therefore, the likelihood of an edge being added to a node cannot be explained purely either by its age or its degree. To address these shortcomings of traditional preferential attachment models, a wide variety of network formation strategies were investigated in Leskovec et al. [2008]. It was shown analytically that the combination of the gap distribution with the node lifetime leads to a power law out-degree distribution that accurately reflects the true network in all four cases.

A discussion of the typical models for group formation in social networks is presented in Backstrom et al. [2006]. The work in Backstrom et al. [2006] studies how the structure and evolution of the communities are related to the network itself. The co-authorship network of DataBase List of Publications (DBLP) was studied, where the conferences serve as proxies for communities. It was shown that the propensity of

[1]The parameter β had different optimal values for different networks.

individuals to join communities and of communities to grow depends on the network structure. Thus, this is closely related to social diffusion studies in the social science community. Specifically, the tendency of an individual to join a community is influenced not just by his or her number of friends within the community, but also by how those friends are connected with one another. An individual is more likely to join a community if the following two hold true:

(1) The number k of friends of the individual in the community is large.
(2) These k friends should be as well linked with one another as possible.

This is consistent with the principle that social diffusion is more likely to occur in highly clustered networks [Centola et al. 2005]. These links correspond to a strong coordination effect and shared focus of interest among group members. It should be pointed out that group formation dynamics are often influenced by factors beyond the structural properties of the network itself. For example, the work in Zheleva et al. [2009] studies the co-evolution of social and affiliation networks and shows that the evolutionary behaviors of these networks strongly influence each other. The work in Zheleva et al. [2009] also proposes a model for understanding the nature of this evolution. This suggests that there are both extrinsic and intrinsic factors to group formation in social networks. This principle has also been studied in Snijders et al. [2007], which provides a model of network evolution in terms of individual behavior.

The typical behavior of large blogs are investigated in Goetz et al. [2009] and McGlohon et al. [2007], and many of these methods can also be used for quantifying the evolution in specific networks. Methods for finding patterns in blog shapes and modeling blog evolution dynamics were discussed in Goetz et al. [2009] and McGlohon et al. [2007]. The work in McGlohon et al. [2007] finds unusual patterns in blog shapes by extracting two sets of features from the topology and the temporal cascade behavior, respectively. A number of interesting properties of blogs were observed in this analysis, and these are described in the application section.

3.1.2. Evolutionary Network Data Generation. An important side effect of the results just describedis that they can be used to create realistic generators of growing networks. This is useful for testing the quality of algorithms for tasks such as community detection. As discussed earlier, evolving networks follow several interesting properties such as Densification Power Law (DPL) and shrinking diameters [Leskovec et al. 2005a]. Leskovec et al. proposed a model [Leskovec et al. 2005a] that can generate graphs over time that satisfy these two properties. The approach is a recursive construction of graph using Kronecker products. The graph at time $t + 1$ is simply the Kronecker product of itself at time t, and it has been shown that such a graph can satisfy the aforementioned properties. The discrete nature of the binary adjacency matrix produces a *staircase effect* in distributions of degree and spectral quantities. The main reason for the staircase effect is that individual values have large multiplicities due to the Kronecker products of binary matrices. To avoid this, it is possible to use probabilistic parameters $0 \leq p \leq r \leq 1$ to generate the adjacency matrix in place of a strictly binary 0–1 matrix. This results in stochastic Kronecker graphs, and they avoid the staircase effect without compromising other desirable properties of deterministic Kronecker graphs.

There are other recursive generative models, such as community-guided attachment and forest fire models [Leskovec et al. 2005b]. The idea behind the community-guided attachment process is to construct communities within communities in recursive fashion. The smallest community is a single node. The recursive structure is simulated by adding nodes in a tree structure at every time point as children of the leaves of the current tree. Let the distance between vertices v and w be denoted by $d(v, w)$ and c

be a constant. Then, independently with probability $c^{-d(v,w)/2}$, the newly added node v connects previously existing nodes in the tree. In the forest fire model, a newly added node connects to randomly chosen x and y number of nodes, where x and y are geometrically distributed with mean $(1-p)^{-1}$ and $(1-rp)^{-1}$ respectively. The parameter p denotes the forward- and r denotes the backward-burning probabilities, and each new node spreads the *fire* via exactly x outgoing edges and y incoming edges that are not yet burnt by the fire, respectively. When there are not enough nodes to burn, the process stops. This generative model is shown to satisfy several properties, such as heavy tailed degree distribution, DPL, and shrinking diameters.

A random graph generator for evolving network has been proposed in Akoglu et al. [2008] using recursive tensor multiplication of an initial matrix with itself up to k times. This simple approach satisfies several interesting properties such as edge weight power law, lambda power law, DPL, shrinking diameters, and many more [Akoglu et al. 2008]. According to the lambda power law, the principal eigenvalues and the number of edges over time follow a power law distribution, $\lambda_1(t) \propto E(t)^{\alpha}$, with the power law exponent α within a certain constant. The key difference between this approach and previous approaches is explicitly capturing the time dimension using a tensor, and the Kronecker recursion is performed over a tensor.

3.1.3. Community Emergence, Evolution, Expansion, and Contraction. Community-based methods are particularly natural to use for evolution analysis because of the ability of clusters to summarize the structure of the network. Therefore, many of the methods proposed for evolutionary clustering can also be used for characterizing the nature of the changes in the data. The main challenge is to create a more tightly integrated framework.

One of the earliest works on community evolution was presented in Hopcroft et al. [2004], who analyzed the Citeseer citation graph from 1990 to 2001. The communities were detected by agglomerative hierarchical clustering, and different snapshots were compared with one another. The communities could be matched to one another between successive snapshots, and their evolution was tracked through time to identify significant structural changes over time, such as the emergence of new communities or the death of old ones. The work in Aggarwal and Yu [2005] explored the expansion and contraction of communities over different snapshots. It constructs a differential graph that measures the changes in the structure of the graph from one snapshot to the next. This is then used to determine expanding and contracting communities.

A method proposed in Palla et al. [2007] extracts communities in each snapshot with the use of the clique percolation method. These communities are then compared with one another over different snapshots to analyze the nature of the underlying evolution. A number of interesting properties about the evolution of small and large communities are observed in this work. It was shown that large groups typically persist longer if they are capable of dynamically altering their membership. In other words, adaptability is a key component of group survival. On the other hand, the opposite is true for small groups, where a smaller amount of change results in greater stability. It was also shown that the knowledge of the time commitment of members to a community can be used to estimate its lifetime. Another work that analyzes the evolution of communities in interaction networks is discussed in Tantipathananandh et al. [2007]. The main observation in this work is that the evolution of communities is gradual and that individuals do not tend to change their "home" community too quickly in most cases. One of the key issues in the effective application of many of the community detection methods is to design ways to "match" the communities over different snapshots in time. This aspect has been studied in detail in Greene et al. [2010].

Another method that uses the group structure of social networks to characterize their evolution is discussed in Berger-Wolf and Saia [2006]. Given a partition

$P^{(t)} = \{g_1^{(t)}, \ldots, g_k^{(t)}\}$ of a vertex set V for every period of observation $t = 1 \ldots T$, this work proposes several computational approaches to understand various dynamics related to evolving groups. The model proposed in this paper constructs a $\beta - graph$ that is a directed acyclic graph (DAG). An edge is added between $g_i^{(t)}$ and $g_j^{(t+1)}$, if the similarity $sim(g_i^{(t)}, g_j^{(t+1)}) \geq \beta$. A MetaGroup (MG) is defined on this DAG as a sequence of groups $\langle g_1^{(i)}, \ldots, g_l^{(j)} \rangle$ in observation interval $[i, j]$, where $(j - i) \leq \alpha$. There are several questions that can be answered in polynomial time using this model. For instance, the number of MGs present in the β-graph is the number of paths of length at least α. This can be computed in polynomial time using dynamic programming. Similarly, the most stable MG is the MG with maximum average edge weight of all the MGs computed. This can be computed in polynomial time using a topologically sorted β-graph. When the group partitions for each snapshot are not given in advance, the problem becomes much harder and has been addressed using recursive enumerations in Ahmed and Karypis [2012].

A tightly integrated framework for clustering and evolution analysis is provided by the *ENetClus* method [Gupta et al. 2011b]. The *ENetClus* method [Gupta et al. 2011b] generalizes the probabilistic *NetClus* [Sun et al. 2009] model to the temporal scenario. This is a soft clustering model that assigns probabilities of membership of each node to different clusters. The idea is to perform the clustering on temporal snapshots of the data. On each snapshot, a probabilistic assignment is learned with the use of the *NetClus* algorithm. The final probabilistic assignment in a given snapshot is used as an initialization point (prior) to the next iteration. This ensures that continuity is maintained among the clusters, and the clusters found in the next snapshot can be directly compared to their counterpart in the current snapshot. A number of evolution metrics are then proposed to measuring significant changes in the cluster behavior. This work shows that temporal community structure analysis exposes several global and local structural properties of networks, which can be quantified in the form of various time series metrics. Examples of such metrics include the cluster membership consistency, cluster novelties, splits, merges, and disappearance. Significant deviations in these values can be reported as anomalous changes in the network.

The work in Gupta et al. [2012b] creates an integrated framework between community detection and matching with the use of an iterative algorithm. This approach sets up an objective function that is based on the matching of the communities between successive snapshots. An objective function is set up to quantify the evolutionary behavior of the communities based on this matching. Such an approach provides evolutionary community outliers that correspond to communities that do not match the communities in the previous snapshot at any significant level. The work in Gupta et al. [2012a] proposes a method for characterizing the "normal" evolutionary behavior of the data. Deviations from this trend are flagged as outliers. Note that this work makes a distinction between "normal" (smoothly evolving) behavior and "abnormal" evolution.

The use of community evolution methods is very common because communities capture the broad patterns in the network. Therefore, a change in the community structure is used to model significant evolution [Malliaros et al. 2012; Sun et al. 2007, 2006, 2008, 2010; Tang et al. 2008]. An overview of methods for performing evolution analysis in networks in the context of the community structure of networks may be found in Spiliopoulou [2011]. A specific and important kind of community-based methods are spectral methods, which is discussed below.

3.1.4. Spectral Methods. Spectral methods are closely related to community detection and are often used to cluster networks [Aggarwal and Reddy 2013]. These methods are also closely related to principal component analysis, although the precise matrix

representation of the network similarity structure or the technique used for principal component analysis may vary with the specific application.

The major advantage of spectral methods is that they use the aggregate correlation structure of the linkages in the network. Such measures are extremely robust to small changes in the underlying network, and a significant change usually reflects a corresponding change in the structure of the network. Although spectral methods can be implemented in a variety of ways, a simple method is to use principal component analysis on its augmented adjacency matrix. Let Q be an $n \times m$ node-link incidence matrix in a network containing n nodes and m edges. This is a binary matrix containing only 0 or 1 values. A value of 1 implies that the corresponding edge is incident on that node. Then, the matrix $A = Q \cdot Q^T$ represents an augmented adjacency matrix, where the diagonal entries are the degrees on the nodes, and all other entries have 0–1 values depending on whether or not a corresponding edge is present. In many interaction networks, weights are naturally associated with the edges. In such cases, the original node-link incidence matrix Q contains the weights instead of unit values. The weighted adjacency matrix $A = Q \cdot Q^T$ can also be defined in a similar way.

The matrix A is guaranteed to be positive semidefinite because this is a property of all matrices of the form $Q \cdot Q^T$. Therefore, the matrix A can be diagonalized as $A = P \cdot D \cdot P^T$. Here, P is an orthonormal matrix whose columns contain the unit eigenvectors of A, and D is a diagonal matrix containing the eigenvalues. The eigenvector corresponding to the largest eigenvalue provides the principal directions of correlation. Significant changes in this vector over the graph snapshots over different periods of time may correspond to anomalous behavior. Such an approach has been used in Idé and Kashima [2004] to determine significant changes in temporally evolving graphs. The principal component is chosen as the activity vector for that graph. This graph is then represented as a time series of activity vectors, which creates a dataset of activity vector values. The principal left singular vector of this dataset provides the significant direction of correlation. The activity vector for the next arriving graph in the series is computed, and the corresponding angle with the principal left singular vector provides the evolution score.

Although the aforementioned method is a simple generalization of principal component analysis, other spectral methods commonly use the Laplacian of the similarity matrix. These methods are more directly related to the communities in the network [Aggarwal and Reddy 2013] and can be used in a similar way. A method for incorporating temporal smoothness in spectral clustering algorithms is discussed in Chi et al. [2009]. Although this method is not designed explicitly for change detection, it can be used as such because an approximate mapping can be found between clusters at different time snapshots. This is because of the incorporation of the temporal smoothness criterion, which allows a clear mapping between clusters at different snapshots. A specific application of this kind of approach to the monitoring of evolution in blog communities is discussed in Ning et al. [2007].

A *compact matrix decomposition* method is proposed in Sun et al. [2007] to approximate the adjacency matrix of large sparse graphs. The primary idea underlying the work is that it is harder to approximate anomalous graphs than normal graphs. Therefore, the approximation error for each graph in a sequence of graphs is constructed. Anomaly detection is performed on this time-series of values.

3.1.5. Shortest Path Distance Evolution. Most real-world graphs such as the Web, social networks, and information networks experience significant changes in terms of the pairwise distances between nodes in the network. For example, it has been shown in Backstrom et al. [2006] that most real graphs such as the Web and social networks

have shrinking diameters over time. This is because edges are continuously added to such networks, which makes them more dense.

In this context, *sudden and abrupt* changes in pairwise distances between nodes are indicative of unusual events in a network. For example, in a bibliographic network such as *DBLP* [Ley 2002], the sudden addition of an edge that connects a pair of widely separated nodes is an unusual event and most likely reflects the sudden collaboration between a pair of authors in different topical areas. Therefore, it is interesting and useful to determine the top-k shortest path distance changes in an evolutionary network. This problem was first proposed in Gupta et al. [2011a].

A straightforward solution to this problem is to solve the all-pairs shortest path problem [Ahuja et al. 1993] at two snapshots, t_1 and t_2. The pairs of nodes for which the distances have changed very significantly are reported. However, such an algorithm requires the (expensive) computation and storage of all-pairs shortest paths, which can be impractical for larger graphs. A key observation in Gupta et al. [2011a] is that edges that lie on the shortest paths between many pairs of nodes in either snapshot are important edges, the addition or deletion of which can significantly change the shortest path distances. Therefore, a randomized algorithm is proposed in Gupta et al. [2011a] to find such edges. This is then leveraged to determine the significant nodes pairs between which the greatest change has occurred. Although the determined node pairs are heuristic in nature, a high amount of precision and recall is achieved by this approach.

3.1.6. Network Evolution with MDL Principle. Methods for monitoring network evolution with the MDL principle are discussed in Ferlez et al. [2008]. Consider a document-word association matrix, where each document has a time-stamp associated with it. The approach constructs snapshots of this association matrix for various time-points. Then the words are clustered in each snapshot using an extension of the standard cross-associations algorithm. Furthermore, the clusters in each snapshot are connected to the neighboring snapshots using the MDL principle. If the clusters did not change significantly in consecutive snapshots, then the snapshots are combined to form a more compact encoding. This reduces the number of unimportant time points and retains only the significant change points over the entire snapshot of the network. The inherent nature of MDL to be parameter-free also becomes an advantage of this model. In addition to finding the change points, this approach also uses the encoding length to detect the emerging and fading clusters.

A method known as *GraphScope* proposed in Sun et al. [2007] is also based on the MDL principle. Intuitively, a change point is one that significantly increases the encoding cost to represent the stream. The approach groups similar sources together into source groups and similar destinations together into destination groups to minimize the encoding cost. If the underlying communities do not change much over time, then the snapshot of the evolving graphs will have similar descriptions and can also be grouped together into a time segment to achieve better compression. Whenever a new graph snapshot cannot fit well into the old segment in terms of this description, *GraphScope* introduces a change point and starts a new segment. This corresponds to a high level of change in the patterns of the underlying network. It has been shown in Sun et al. [2007] that such change points correspond to drastic discontinuities in the network. Readers are referred to Sun et al. [2007] for details.

3.1.7. Role Dynamics for Understanding Network Evolution. The nodes in most social and information networks are often associated with roles, which may dynamically evolve along with the network structure over time. Therefore, an interesting perspective in network evolution is to understand the underlying role dynamics [Rossi et al. 2012]. For example, a node could be in a center of a star network, or it could be a broker transferring information between two different communities. Understanding such

individual node characteristics can help us in understanding global network processes such as homophily. The role dynamics approach [Rossi et al. 2012] can also be used to find outliers, where a node transitions multiple roles within a short time period at an uncommon rate. Also role statistics can be used to find similarity between evolving networks and is extremely useful in validating synthetic network generators.

One approach discussed in Rossi et al. [2012] consumes snapshots of the graph adjacency matrix $\mathbf{A}_t$ for each time point t. Then, a feature extraction approach is applied on each $\mathbf{A}_t$ to generate a node-feature matrix $\mathbf{V}_t$. These features are variants of degree and ego-network measures and represent local, community-level, and global properties of a node in the network. Non-negative Matrix Factorization (NMF) is applied to the extracted features, with the MDL criterion. The NMF minimizes the following squared error term, while the low rank factor for NMF is chosen by MDL:

$$\min \frac{1}{2} \left\| \mathbf{V}_t - \mathbf{G}_t \mathbf{F} \right\|_F^2 . \tag{7}$$

The rank of the matrix $\mathbf{G}_t$, which is the number of representative roles, signifies the model complexity, and MDL chooses the optimum model complexity without compromising much on the model quality. The learned role-feature matrix $\mathbf{F}$ represents the contribution of each role on extracted features. The factors can then be used to analyze the role dynamics, such as role importance. One can measure the importance of each role over time using $\mathbf{G}_t^T e/n_t$, where e is a vector of ones and n_t is the number of nodes at time t. The activity of roles during different periods of time can be used to understand the effect of roles over time; for instance, a coordinator at work may be extremely active during the day but show no activity during the night. Also some roles are found to have complete inactivity, and a sudden uprise of activity in these roles marks a beginning, change, or end of a new event. Some roles may decrease/increase in importance depending on the period of the underlying event. For example, at the beginning of a conference, users may exchange message to only a few known people, and the graph may be relatively sparse; but by the end of the conference, they may be well connected, resulting in an overall higher number of betweenness nodes.

3.1.8. Visualizing Evolutionary Networks. An important way of understanding the nature of evolution of social networks is with the use of visual analysis [Brandes and Corman 2003; Chen and Morris 2003; Falkowski et al. 2006; Sallaberry et al. 2013; Chen 2006; Moody et al. 2005; Bender-deMoll and McFarland 2006]. One of the earliest methods discussed [Chen and Morris 2003] uses reduced representations of the underlying network to understand the nature of the changes. Two widely known link reduction algorithms, known as minimum spanning trees (MSTs) and Pathfinder networks (PFNETs), are used to model the evolution of the underlying network. These two methods are compared in Chen and Morris [2003] in terms of their effectiveness on scientific co-citation networks. It has been suggested that PFNET models are generally superior to MST-based models because the latter models are focused mostly on high-degree nodes, which are often inadequate to explain the underlying network. On the other hand, PFNET models provide a more intuitive explanation of the underlying evolution paths.

The work in Brandes and Corman [2003] focuses on networks of dynamic discourse that evolves over time. The nodes in this network are made of nouns and adjectives, and an edge represents the co-occurence of these entities in a sentence. Such networks are important in social science in understanding the evolving patterns of conversations over time. The work in Brandes and Corman [2003] introduces a method for visualizing such networks, but it can also be applied to other kinds of network. A state-based approach is used to model the evolution. In addition to the intermediate states of the

network, it conveys the nature of change between states by unrolling the dynamics of the network. Each modification is shown in a separate layer of a three-dimensional representation, where the stack of layers corresponds to a time line of the evolution.

Because the community detection problem is closely related to all forms of evolution analysis, it is natural to design a method that can integrate community analysis with visualization. The work in Falkowski et al. [2006] is one of the early works along this line, which integrates community detection with the visualization problem. The work in Sallaberry et al. [2013] integrates the clustering problem with that of visual analysis in evolving networks. This is because visual representations provide excellent summary insights into the underlying network.

Visual analysis is particularly interesting when it is performed in the context of specific applications, where the evolution behavior is easy to interpret. For example, the Web continuously evolves over time, which leads to significant changes in the distribution of pages over different sites. A study of the evolution of different Web ecologies with the use of visual analysis is provided in Chi et al. [1998].

3.1.9. Outlier Detection. Many forms of pattern changes in a network may be characterized in the form of evolution rules. In the framework presented in Berlingerio et al. [2009b], nodes and edges have labels associated with specific properties of the network. Furthermore, edges contain the time-stamps corresponding to their first appearance. Patterns are defined as subgraphs, which have similar structure and labels on nodes at different time-stamps, and the same relative offsets of the time-stamps. This defines significant temporal patterns or *graph evolution rules* in the underlying data. Evolution rules do not necessarily represent outliers because they correspond to frequent temporal patterns in the data. On the other hand, the formation of a new evolution rule at a given time may be considered a temporal novelty and may be reported as an outlier.

Evolution analysis can be defined in an almost unlimited number of ways in temporal graphs because of the different combinations of time and structure, which can be used to define regularity. Some of the earliest work focuses on measuring similarities between successive snapshots of graphs with the use of different similarity functions [Papadimitriou et al. 2010; Pincombe 2005; Shoubridge et al. 2002]. Another method that uses graph matching between successive snapshots for anomaly detection is discussed in Showbridge et al. [1999]. This creates a time-series that can be analyzed with standard autoregressive moving average (ARMA) methods for finding the outliers. In the context of similarity-based measures, a large number of possibilities are available in terms of how similarity is computed between different snapshots. They could be based on eigenvalues, entropy, network topology, or node or edge properties [Akoglu and Faloutsos 2013]. For instance, the spectral distance between two graphs is proportional to the sum of squared differences of eigenvalues of the Laplacian. Formally, the spectral distance between two graph instances G and H is defined here, where λ_i and μ_i are the eigenvalues of the Laplacians of G and H, respectively:

$$d(G,H)^2 = \sum_{i=1}^{k} (\lambda_i - \mu_i)^2 / \min\left\{\sum_i \lambda_i^2, \sum_j \mu_j^2\right\}.$$

The work in Priebe et al. [2005] uses the history of a node's neighborhood to detect anomalies. Some of these methods [Sun et al. 2007, 2008] are specifically applicable to bipartite graphs. The determination of significant evolution in graphs can be useful in the context of a wide variety of applications such as monitoring blog communities [Ning et al. 2007] or mining traffic flow datasets [Mongiovi et al. 2013]. In the latter case, values are associated with edges corresponding to traffic flows. Anomalous regions are found in the network by using the values on these edges.

3.2. Streaming Scenario

The problem of determination of unusual objects (or temporal outliers) is discussed in Aggarwal et al. [2011]. Consider a partitioning of the nodes denoted by $\mathcal{C} = C_1 \ldots C_{k(\mathcal{C})}$. The number of node partitions in $\mathcal{C}$ is denoted by $k(\mathcal{C})$. Each set C_i represents a disjoint subset of the nodes in V. The likelihood fit for an edge is defined as its probability of presence based on a generative model. For example, an edge between two co-authors from very different communities in a bibliographic network would have a very low fit value. Edges and subgraphs are quantified using this fit value and reported as anomalies. Key evolutionary changes are reported as temporal outliers.

To enable the aforementioned analysis, cluster-based partitions $\mathcal{C} = C_1 \ldots C_{k(\mathcal{C})}$ need to be maintained dynamically from the edge stream to perform the linkage anomaly detection. It is well known that the use of edge sampling [Karger 2000] can be used to create dense partitions. For example, a sample of edges from a stream implicitly creates a set of clusters in terms of the connected components in this sample. Such connected components are much denser than randomly picked node sets in the graph because of the inherent bias of edge sampling [Karger 2000]. A major challenge arises in adapting the minimum 2-way cut methods of Karger [2000] to a more general stream scenario while maintaining specific *structural properties* of the k-way cut partitions. For example, one possible structural constraint would be to ensure a minimum number of points in each cluster or to constrain the total number of clusters. Clearly, a random edge sample may not satisfy such constraints. Reservoir sampling [Vitter 1985] is a methodology to dynamically maintain an unbiased sample from a stream of elements. The method of Aggarwal et al. [2011] extends this method to an unbiased sample of a structured graph, so that many natural and desirable structural properties of the sample are maintained. This goal is achieved with the help of a *monotonic set function* of the underlying edges in the reservoir. A monotonic set function is defined on the sample as follows.

Definition 3.1 *(Monotonic Set Function).* A monotonically nondecreasing (nonincreasing) set function is a function $f(\cdot)$ whose argument is a set, and whose value is a real number that always satisfies the following property:

—If S_1 is a superset (subset) of S_2, then $f(S_1) \geq f(S_2)$.

The monotonic set function can be useful for regulating the structural characteristics of the graph over a given set of edges. Some examples of a monotonic set function include the number of connected components in the edge set S (monotonically nonincreasing) or the number of nodes in the largest connected component in edge set S (monotonically nondecreasing). Properties such as these are very useful for inducing the appropriate partitions with robust structural behavior. In some cases, it is possible to use *thresholds* on these properties, which are also referred to as *stochastic stopping criteria*. It has been shown in Aggarwal et al. [2011] that thresholds on these stopping criteria can be translated to thresholds on a hash function that is applied to the edges. This is used to create a reservoir sampling algorithm, which uses a hash-based algorithm to perform admission control in the reservoir of edges. This has been used to maintain the partitioning continuously and report edges in the network that have very low likelihood fit. It should be pointed out that such an approach is able to continuously maintain both the clusters *and* also detect important evolutionary edges in the network over time.

The work in Yu et al. [2013] defines a different notion of outliers, where unusual changes in the neighborhood of a node are discovered and reported in a graph stream. These unusual changes could be defined either in terms of the *level* of activity or the *patterns* of activity. It has been shown in Yu et al. [2013] that an eigenvector-based

approach can be used, where the eigenvectors of the subgraph in the neighborhood of a node are determined. Changes in the eigenvalues represent activity-level changes, whereas changes in the eigenvector directions represent changes in neighborhood subgraph patterns.

The problem of influence analysis has been studied in evolving network streams [Aggarwal et al. 2012a]. Many social networks may be defined in the form of *transient interactions* between entities. In such cases, edges may be rapidly added to and deleted from the network, as a result of which the topology of the network may vary drastically over time. Many natural social interactions, such as epidemiological networks, email networks, or chat networks can be modeled much more naturally using this approach. A stochastic approach was proposed in Aggarwal et al. [2012a] to determine the information flow authorities with the use of a globally optimized forward trace approach and a locally optimized backward approach. The key idea is that the flows in the network and the changes in network structure are both analyzed in parallel. Therefore, the flow variables in the network are time-stamped, and the values at time $(t+1)$ can be derived from those at time t by using the network structure at time t. A greedy approach is developed in which new nodes are added to or deleted from the current set of influence points to improve the global influence objective function. The approach has also been generalized to the case of social streams in Subbian et al. [2013].

4. INCORPORATING CONTENT IN EVOLUTION ANALYSIS

Content provides unprecedented scenarios for analysis because it can be used to make more informed inferences about the underlying data. In many cases, content and structure evolves simultaneously, and the dynamics of the evolution can be related between the two aspects.

One of the most common scenarios for evolution analysis is a *social stream* in which the streams of content created by different users have both structure and content. For example, each *Twitter* post contains the content of the tweet, as well as the set of users (followers or network actors) to whom this tweet is sent. In such cases, it is useful to determine key events from the underlying social stream by combining the information available in the content and the structure. The work in Aggarwal and Subbian [2012] defines such a model in which structure and content are used to determine key events from the social stream. A clustering approach is used to summarize the social stream, and the evolution in the underlying clusters is used to detect events in the social stream. A supervised approach is also used to detect events more accurately when previous examples of rare events are available. The broad approach in this work uses two phases:

(1) In the first phase, clusters are continuously maintained as the social stream is received over time. The similarity function uses both the content and the structure to do a partition-based clustering of the stream.
(2) In the second phase, the clusters are leveraged to discover events from the social stream. These events could be discovered either in a supervised or unsupervised manner, depending on whether or not ground-truth events are available.

One challenge with the use of the approach are the high computational and memory overheads in maintaining the information about the content and structure of the different clusters. Therefore, a sketch-based approach [Aggarwal and Yu 2007] is used to compress the structural and content representation of the underlying social stream. Other methods for event detection in a variety of social streams are discussed in Sakaki et al. [2010], Lin et al. [2010], Sayyadi et al. [2009], and Zhao et al. [2007]. In the context of clustering problems, the most common scenario analyzed is that of blogs [McGlohon et al. 2007; Goetz et al. 2009; Ning et al. 2007], where the content of the blog influences

its linkage structure. This is discussed in some detail in the application section. Numerous applications in mobile networks [De Melo et al. 2010] also use meta-information, such as call-duration, to analyze the underlying network evolution. A content- and network-based flow mining approach for dynamic influence analysis was also proposed in Subbian et al. [2013]. In this case, sequential patterns are dynamically mined from a combination of the keywords and the dynamic network in the social stream. These are then used to predict the most influential entities in a dynamic and evolving network.

The problem of classification is addressed in Aggarwal and Li [2011], in which content and structure are combined for the problem of dynamic classification. In this work, a random walk approach is used to create a classification model. The original network $G = (N, A)$ contains a node set N and edge set A, such that each node in N has a set of keywords associated with it. Each of these keywords is converted into a new pseudo-node to create an augmented network $G' = (N \cup N', A \cup A')$. An edge is added from a node in N' to a node in N when the corresponding keyword is present in that node. This corresponds to the newly added edges in A'. This results in a semi-bipartite network. A random walk approach is used to perform the classification. When a random walk is performed a node in N, the majority label of the nodes visited is reported as the relevant class label. The approach works for dynamic evolving networks as well because the semi-bipartite representation and associated index structures are maintained dynamically for fast processing. More details of the dynamic maintenance may be found in Aggarwal and Li [2011].

The problem of link prediction has also been studied in the context of dynamic networks with content [Aggarwal et al. 2012b]. The work in Aggarwal et al. [2012b] uses a dynamic clustering approach, wherein a rough clustering of the network is maintained continuously. This rough clustering is based only on the structure, and it provides the macro-clusters over which more fine-grained analysis is performed to predict the underlying links. The underlying links are predicted with the use of a combination of the content and structure within each region of the network. The approach has been shown to be significantly superior to many traditional methods for link prediction and is also applicable to heterogeneous network scenario.

A method for performing graph stream clustering with side information is discussed in Zhao and Yu [2013]. Such side information is often defined by the underlying content and can be very useful in many scenarios. For example, in social networks, user profiles and behaviors can be used as side information. In Web click graphs, the meta-information about the user Web pages can be utilized, and in bibliographic networks, the information about the underlying publication can be used as side information. It has been shown in Zhao and Yu [2013] that such side information can be used to significantly improve the clustering process. The approach described in this work is an extension of the technique proposed in Aggarwal et al. [2010]. The method in Zhao and Yu [2013] combines structural and content-based distances to perform the clustering. As in Aggarwal et al. [2010], a sketch-based approach is used to address the high memory requirements.

5. APPLICATIONS

In this section, we discuss numerous applications of evolutionary network analysis. The focus is on how the modeling is done, rather than the specific details of the methodology for each application. It will be evident from the discussion of this section that evolutionary network analysis is useful for a very wide variety of domains such as social networks, blogs, or road networks. An overview of the key applications are summarized in Table II. A broad discussion of different kinds of evolution analysis in the context of different kinds of networks may be found in Akoglu and Faloutsos [2013] and Dorogovtsev and Mendes [2003].

Table II. List of Key Applications of Evolutionary Network Analysis

Domain	Work
World Wide Web	[Dorogovtsev and Mendes 2003] [Papadimitriou et al. 2010] [Chan et al. 2008] [Chi et al. 1998]
Telecommunication Networks	[Liu et al. 2011] [De Melo et al. 2010] [Akoglu and Faloutsos 2010] [Akoglu and Dalvi 2010]
Communication Networks	[Chan et al. 2008] [Huang and Lin 2009]
Road Networks	[Mongiovi et al. 2013] [Bogdanov et al. 2011]
Recommendations	[Huang et al. 2005] [Aggarwal et al. 2012b] [Leskovec et al. 2007] [Aggarwal et al. 2012a] [Richardson and Domingos 2002] [Sarkar et al. 2012] [Tylenda et al. 2009] [Huang and Lin 2009]
Social Network Events	[Aggarwal and Subbian 2012] [Sayyadi et al. 2009] [Sakaki et al. 2010] [Zhao et al. 2007] [Silva and Willett 2008] [Tong et al. 2008c] [Lin et al. 2010] [Beutel et al. 2013]
Blog Evolution	[Ning et al. 2007] [McGlohon et al. 2007] [Goetz et al. 2009] [Leskovec et al. 2007]
Computer Systems	[Idé and Kashima 2004] [Albert et al. 2000]
News Networks	[Yan et al. 2012] [Leskovec et al. 2009]
Bibliographic Networks	[Gupta et al. 2011b] [Chen 2006] [Sun et al. 2011] [Barabâsi et al. 2002]
Biological Networks	[Vázquez et al. 2002] [Solé et al. 2002] [Asur et al. 2007] [Dorogovtsev and Mendes 2003] [Teichmann and Babu 2004] [Stuart et al. 2003] [Beyer et al. 2010]

5.1. World Wide Web

Web graphs are approximate snapshots of the Web created by search engines. Such approximate snapshots are used to answer search engine queries in which the PageRank is computed from the structure of the Web graph. By continuously monitoring the changes, it is possible to determine the amount and significance of changes in the Web. Such measures provide insights into the robustness of the content acquired from the Web. A Web host that is unavailable at crawl time may cause the crawler to miss the content from that site. Because the crawl typically occurs over multiple days, during which time the Web structure may change, this may sometimes even lead to invalid or corrupt data. In this context, a useful approach is to compare the snapshots of Web structure at different time instances to identify anomalies. For example, if a group of IP addresses are missing during the acquiring of the content, this will also be reflected in the corresponding Web graph at that snapshot with respect to the previous snapshot. The work in Papadimitriou et al. [2010] proposes five similarity schemes for measuring the similarities between the different graph snapshots. Three of these are adapted from existing similarity measures [Bunke et al. 2006], whereas the other two are the shingling and random projection methods. The latter pair are adapted from document and vector similarity measures. The idea is to identify anomalies that occur in the crawling process as a result of hardware or other problems. In many cases, it is also desirable to understand the evolution of the Web over time with visual representations. A study of the evolution of different Web ecologies with the use of visual analysis is provided in Chi et al. [1998]. Another common application of network evolution is the analysis of user click streams on the Web or query-click pairs from search engine logs, which can be represented as evolving graphs. Because external events often influence

the user click behavior significantly, Chan et al. [2008] proposed methods to identify such external events.

5.2. Telecommunication and Mobile Networks

Telecommunication and mobile networks can be modeled as nodes corresponding to the different participants, with the edges representing either the network connections or the interactions between the different participants. This provides a wide variety of applications that can be modeled in this context. The work in Liu et al. [2011] identifies important features of the mobile phone graph at any point in time and provides ways in which to model these features in an interpretable way. For this purpose, the work analyzes a massive who-calls-whom network for as long as a year and gathers records of two large mobile phone communication networks with 2 million users and 2 billion calls. The calling behavior distribution was analyzed at multiple time scales, and it was shown that the distribution is skewed, with a heavy tail that changes at different time scales. The concept of a δ-stable distribution is defined in the context of a multiscale distribution fitting problem. A framework, *ScalePower,* is proposed to analyze the distribution at different time scales. It is shown that this framework fits the multiscale data distribution very well and provides explanatory insights.

The evolution of the interactions in mobile networks often corresponds to important network events because events such as festivals have an effect on the interactions of individuals. The work in Akoglu and Dalvi [2010] observes that important structural properties of the network, such as the neighborhood overlap and clustering coefficient, influence the tie strengths and link persistence between individuals. Furthermore, a change-point detection method is proposed for analyzing user behaviors with the use of eigenvalue analysis. It was shown that these change points often correspond to important social events and festivals in the data. The work in Akoglu and Faloutsos [2010] proposes an algorithm that operates on a time-varying network of agents, in which edges represent the interactions between the different individuals. The algorithm is designed to determine anomalous points in time, in which agents change their behavior significantly. The algorithm also determines the attributes that contribute to most of the changes. Methods for finding surprising patterns in the call duration of mobile phone users are discussed in De Melo et al. [2010].

5.3. Communication Networks

The work in Chan et al. [2008] focuses on the problem of finding correlated spatiotemporal changes in large communication networks. When a fault occurs in a communications network, it typically induces changes in the routing topology of the network. For example, when an IP router fails, all paths that pass through that router will also not be available for communication. To find root causes of the failure in such networks, traditional methods such as active probing may often be too expensive. Therefore, a natural approach would be to use the changes in the end-to-end routes to determine faults. The idea is to partition the changes based on spatial (topological) locality and temporal locality. Each such group is often more likely to be caused by a single fault. The work in Chan et al. [2008] uses the regions of correlated spatiotemporal change to identify the root cause of communication network faults. The detection of repeated links between nodes in communications nodes has also been used for communication network surveillance [Huang and Lin 2009].

5.4. Road Networks

Evolving network analysis is important in the context of road networks. For example, the work in Mongiovi et al. [2013] models the set of roads as a network and the traffic on the roads as values on the corresponding edges. Thus, the evolution is measured

in terms of the changes in these values rather than in the network itself. Thus, the problem is one of content evolution in the context of a base network. Clearly, significant anomalous regions of change may correspond to traffic events in the underlying data.

A related problem is that of mining *heavy subgraphs* in time-evolving networks [Bogdanov et al. 2011]. These heavy subgraphs correspond to regions in the network in which the values on the edges are high in localized regions over time. Clearly, these correspond to the high-traffic regions in the data. The work in Bogdanov et al. [2011] shows that the problem of finding the heaviest dynamic subgraph is NP-hard. An algorithm known as *MEDEN* is proposed in Bogdanov et al. [2011], and it shows that the algorithm is able to find regions of congestion in a large road network from Los Angeles. It has been shown in Bogdanov et al. [2011] that the applicability of this approach is quite general, and it can be used for social networks rather than road networks.

5.5. Social Network Recommendations

The problem of link prediction [Liben-Nowell and Kleinberg 2007; Sarukkai 2000; Taskar et al. 2003; Al Hasan et al. 2006; Popescul and Ungar 2003] is used directly in social networks to suggest friends for different users. This broad approach can also be used for product recommendations in social media networks that are richer in terms of the level of content available. Some of the recent work [Aggarwal et al. 2012b; Backstrom and Leskovec 2011] also uses the content in the network for better link prediction, whereas the work in Aggarwal et al. [2012b], Sun et al. [2012], Kolar et al. [2010], Huang and Lin [2009], Tylenda et al. [2009], and Sarkar et al. [2012] uses the temporal component of link prediction more explicitly in terms of links arriving at different moments in time rather than a single snapshot. The use of content is particularly useful for social media and product networks, where other forms of media, such as text, images, or video, are often available with the network structure. The relationship between the link recommendation problem and collaborative filtering problem has been explicitly explored in Huang et al. [2005]. The idea is to use a graph-based approach to model transitive user-item associations. Once the graph model has been constructed, many off-the-shelf network analysis and link prediction methods can be used for making recommendations. The advantage of this approach over traditional collaborative filtering approaches is the richness of the network representation and the wide variety of network analysis measures that can be used for making more effective recommendations.

Another method for making recommendations in social networks are techniques designed for influence analysis [Kempe et al. 2003]. These techniques recommend customers to the merchant (in the context of a network) rather than products to customers (as in traditional collaborative filtering). The core idea here is that customers frequently interact with one another and influence each other. Therefore, by picking a few well-chosen customers, it is often possible to have outsized influence in the recommendation process. Thus, at the end of the day, the goal is to perform the product recommendations in a viral manner though the customer network [Richardson and Domingos 2002]. However, much of this work is performed in the context of static networks, although some recent work has also extended it to dynamic networks with transient interactions [Aggarwal et al. 2012a].

The analysis of cascading behavior in large networks is an interesting problem that examines the propagation of content in the network [Leskovec et al. 2007; Prakash and Faloutsos 2012; Subbian and Melville 2011]. The blog posts influence the posts by other users and also change the inherent evolution of the links. Such evolution is particularly common in the context of important external events. The blogs create a publicly available record of how information is propagated in the network. The work in

Leskovec et al. [2007] presents a simple model that mimics the spread of information on the blogosphere and produces information cascades very similar to those found in real life. The study of such influence patterns in the context of news networks is provided in Yan et al. [2012] and Leskovec et al. [2009]. Influence analysis [Kempe et al. 2003; Leskovec et al. 2007; Richardson and Domingos 2002] is used to make recommendations to customers in networks. However, in most of these cases, it is assumed that the dynamics is in the influence pattern, whereas the network itself is fixed.

5.6. Social Network Event Detection

A wide variety of events are of interest in social networks, such as unusual tweets, meetings, or changes in trends in the content of the underlying network. Social networks often result in large volumes of tweets, which are referred to as *social streams*. The work in Aggarwal and Subbian [2012] examines the problem of event detection in the context of social streams by examining the changes in both the content and the structure of the underlying social stream. Both supervised and unsupervised models are proposed for event detection. Other models for event detection in social streams such as *Twitter* are discussed in Sakaki et al. [2010], Lin et al. [2010], Sayyadi et al. [2009], and Zhao et al. [2007].

The work in Silva and Willett [2008] determines anomalous meetings in social networks by using the recorded meetings. The level of the anomaly is also explicitly quantified by the level of evolution from the previously recorded meetings. The main challenge in this problem is that the number of observed meetings is much smaller than the number of nodes in the social network. The work in Silva and Willett [2008] uses a hypergraph setting, in which edges are used to connect more than two vertices simultaneously. The distribution of meetings was modeled as a two-component mixture of a "nominal" distribution and a distribution of anomalous events. A variational EM-approach was used to assess the likelihood of each observation being anomalous. A somewhat more complex version of the problem is proposed in Tong et al. [2008c], where an anomalous event could either be a meeting or a publication with its associated set of content such as keywords. It has been shown [Tong et al. 2008c] that the transformation of this problem to an anomaly detection problem is advantageous because it brings the vast analytical of graph analysis into play. This is generally the case for many network analysis applications because of the inherent richness in the graph representation structure.

An important event in many Web services, one that depend on user-generated content, is the positing of fraudulent input by spammers. For example, in the context of a social network such as *Facebook*, one can try to discern the set of fraudulently obtained page likes. The work in Beutel et al. [2013] proposes *CopyCatch* that determines lockstep page like patterns on *Facebook* by analyzing the social graph between users and pages and the times at which they were created. The suspicious behavior is modeled in terms of graph structure and edge constraints. Two algorithms were proposed, one of which is highly scalable with the use of a *MapReduce* implementation. The method in Beutel et al. [2013] was shown to severely limit the greedy attacks in very large user networks such as *Facebook*. It was also suggested [Beutel et al. 2013] that the problem has potential extensions to event detection in other related social network analysis domains.

5.7. Computer Systems

A method for (evolutionary) anomaly detection in computer systems with the use of analytical modeling is discussed in Idé and Kashima [2004]. The idea is that unusual evolutions in the dependency graph are interesting as anomalies and should be investigated further. The approach is an automated run-time anomaly detection method

at the application layer of a multimodal computer system. The first step is to model a Web-based system as an evolving network. Specifically, the Web-based system is modeled as a weighted graph in which each node represents a service and each edge represents a dependency between services. Note that these dependencies may vary significantly with time. The patterns of changes are determined in Idé and Kashima [2004] by performing dynamic eigenvector analysis on the dependency structure of the graph. The structural analysis of attacks in complex networks are discussed in Albert et al. [2000].

5.8. Blog Evolution

Blogs can be considered rich evolving networks, where each blog post is a node, and a hyperlink between two blog posts can be considered an edge. Blogs evolve relatively fast because of the open nature of such publishing, and their evolution is often a direct result of important events in the external world. For example, a significant event such as a hurricane may impact the evolution of a blog very differently than another event, such as an election. In this context, the evolution of the structure of blogs provides important insights about the nature of the underlying events. A spectral method for modeling blog evolution was proposed in Ning et al. [2007]. The work in Goetz et al. [2009] creates an explicit model of blog dynamics, whereas that in McGlohon et al. [2007] determines patterns in blog shapes. The work in McGlohon et al. [2007] finds unusual patterns in blog shapes by extracting two sets of features from the topology and the temporal cascade behavior, respectively. It was shown that the topology features can help in distinguishing between different blog subjects such as "*humor*" or "*conservative*." It was also shown that the temporal activity of blogs is very nonuniform and bursty, is often self-similar, and can be characterized by a *bias factor*. The work in Leskovec et al. [2007] provides a simple model of how information is propagated in cascades over the blogosphere. This is useful in understanding how information flows in the blogosphere may happen.

5.9. News Networks

A very closely related model to blog networks is that of news networks [Yan et al. 2012]. A significance-driven framework was proposed in Yan et al. [2012] to characterize the evolution of local topology and find dynamic patterns with evidently statistical significance for temporally varying news report networks. Two quantifications, which are referred to as the *potential index* and the *evolving score,* were proposed for evaluating evolving patterns. A systematic analysis is provided in Yan et al. [2012] for one real news network with these quantifications. It was shown that the method proposed can effectively find the evolving characteristics and extract significant dynamic patterns from news networks. It should be pointed out that many of the aforementioned techniques that were developed originally for blog networks are applicable to news networks as well. The evolution of "hot ideas" over news networks, which are also referred to as "memes" [Leskovec et al. 2009], is also important from the perspective of identifying influential news stories and their evolution. Although the work in Leskovec et al. [2009] is independent of network structure, a significant potential exists in terms of relating the evolution of such news stories to the structural behavior of blog patterns.

5.10. Bibliographic Networks

Bibliographic networks are a particularly popular benchmark for a significant number of social network analysis papers. Most of the social network analysis algorithms are tested on such data because of the relatively clean nature of such datasets. However, a number of methods are explicitly tailored to such networks. Bibliographic networks can be especially challenging when a *heterogeneous* representation is used in which

the authors, conferences, or keywords are treated as different types of nodes that are connected to one another with different kinds of links. In cases where the authors are connected, the evolution semantics relate to evolution of social connections [Barabâsi et al. 2002; Sun et al. 2011]. However, if the links also connect authors and conferences/keywords [Gupta et al. 2011b], the semantics of the network provides insights into how the topical area of the authors evolves with time. Networks that are combined across heterogeneous types of connections provide a broader perspective into the evolution of different kinds of connections [Gupta et al. 2011b; Sun et al. 2011]. Methods for visualization of bibliographic networks are discussed in Chen [2006].

5.11. Biological Networks

In protein-protein interaction networks Vázquez et al. [2002] and Solé et al. [2002], nodes correspond to proteins and edges correspond to interactions between them. The interactions between the networks often change with the age of the protein, and the changes in interaction have a direct impact on diseases. Another example is the case of metabolic pathway networks, in which nodes correspond to intermediate metabolic products, and the edges correspond to transformations between them. In several diseases, such as type-2 diabetes, the disruption of insulin-related metabolic pathways leads to the evolution of these networks [Beyer et al. 2010]. Therefore, if the specific evolution patterns of the networks in individuals can be captured over time, it can lead to diagnostic insights about the nature of the changes in such networks.

Individual gene expressions can be modeled as networks in which the variation in the gene expressions of an individual can be captured by a temporal network. A node in such a network is a gene. When two genes have a similar change in their expression over a short period because of external or internal factors (e.g., drug administration to a cancer patient), an edge is added between them. Thus, the network captures similarity in gene expression. The long-term evolution of these networks provides an understanding of how the correlations between communities of genes are impacted by various external and internal factors [Stuart et al. 2003]. The work in Asur et al. [2007] showed how a patient-patient correlation network can be used to measure the clinical impact of drug toxicity on patients. In this case, the nodes correspond to patients, and the edges correspond to the similarity in their reactions to a particular drug. The evolving communities in this network provide insights into groups of patients who are impacted in a similar way. A discussion of several aspects of biological evolution analysis may be found in Dorogovtsev and Mendes [2003] and Teichmann and Babu [2004].

6. CONCLUSIONS

This article provides an overview of the key methods used for evolution analysis of dynamic graphs. This includes both methods for maintenance and methods for evolution analysis of the underlying graphs. Both the snapshot and streaming scenario were discussed in this survey. The latter scenario is significantly more challenging from a computational perspective. Methods for incorporating content in the evolution analysis process were also discussed. The applications of evolutionary analysis are quite diverse and were discussed in detail.

There is significant scope for future research in evolutionary network analysis. The area of streaming is still relatively new, and the techniques are being generalized to many newer problems. The streaming scenario also presents numerous challenges because of the challenges in maintaining real-time structural summaries. This is a significant area of future research in several social network analysis areas, such as link prediction and social influence analysis. The work in content-centric analysis is also relatively limited. Most of the work on content-centric analysis is designed for static

networks, and many methods such as collective classification have not been generalized to streaming networks. When content is available with the evolving network, the associated challenges become much more significant because of the co-evolution of the content with network structure. A related area that combines content-centric analysis and the streaming scenario is that of *social streams*, such as *Twitter* streams, in which structure is dynamically combined with content. Finally, while the applications of network evolution analysis are diverse, we have barely scratched the surface of the vast number of problems and domains in which evolution analysis can be leveraged.

ACKNOWLEDGMENTS

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. 2009. Link prediction on evolving data using matrix and tensor factorizations. *ICDMW*. IEEE, 262–269.

Charu Aggarwal. 2011. On classification of graph streams. *SDM*, 652–663.

Charu Aggarwal (Ed.). 2011. *Social Network Data Analytics*. Springer.

Charu Aggarwal and Nan Li. 2011. On node classification in dynamic content-based networks. *SDM*, 355–366.

Charu Aggarwal, Yao Li, Philip Yu, and Ruoming Jin. 2010. On dense pattern mining in graph streams. *PVLDB* 3, 1 (2010), 975–984.

Charu Aggarwal, Shuyang Lin, and Philip Yu. 2012a. On influential node discovery in dynamic social networks. *SDM*, 636–647.

Charu Aggarwal and Chandan K. Reddy. 2013. *Data Clustering: Algorithms and Applications*. CRC Press.

Charu Aggarwal and Karthik Subbian. 2012. Event detection in social streams. *SDM*, 624–635.

Charu Aggarwal, Yan Xie, and Philip Yu. 2012b. On dynamic link inference in heterogeneous networks. *SDM*, 415–426.

Charu Aggarwal and Philip Yu. 2005. Online analysis of community evolution in data streams. *SDM*, 56–67.

Charu Aggarwal, Yuchen Zhao, and Philip Yu. 2010. On clustering graph streams. *SDM*, 478–489.

Charu Aggarwal, Yuchen Zhao, and Philip Yu. 2011. Outlier detection in graph streams. *ICDE*, 399–409.

Charu C. Aggarwal and Haixun Wang. 2010. *Managing and Mining Graph Data*. Vol. 40. Springer.

Charu C. Aggarwal and Philip Yu. 2007. *A Survey of Synopsis Construction in Data Streams*. *Data Streams*. Springer, 169–207.

Rezwan Ahmed and George Karypis. 2012. Algorithms for mining the evolution of conserved relational states in dynamic networks. *Knowledge and Information Systems* 33, 3 (2012), 603–630.

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. 1993. *Network Flows—Theory, Algorithms and Applications*. Prentice Hall, I–XV, 1–846 pages.

Leman Akoglu and Bhavana Dalvi. 2010. Structure, tie persistence and event detection in large phone and SMS networks. *MLG Workshop*. ACM, 10–17.

Leman Akoglu and Christos Faloutsos. 2010. Event detection in time series of mobile communication graphs. *Army Science Conference*.

Leman Akoglu and Christos Faloutsos. 2013. Anomaly, event, and fraud detection in large network datasets. *WSDM*, 773–774.

Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2008. RTM: Laws and a recursive generator for weighted time-evolving graphs. *ICDM*, 701–706.

Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. 2006. Link prediction using supervised learning. *SDM Workshop on Link Analysis, Counter-terrorism and Security*.

Réka Albert and Albert-László Barabási. 2000. Topology of evolving networks: Local events and universality. *Physical Review Letters* 85, 24 (2000), 5234–5237.

Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 1 (2002), 47.

Réka Albert, Hawoong Jeong, and Albert-László Barabási. 2000. Error and attack tolerance of complex networks. *Nature* 406, 6794 (2000), 378–382.

Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. 2007. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *KDD*. ACM, 913–921.

Lars Backstrom, Daniel P. Huttenlocher, Jon M. Kleinberg, and Xiangyang Lan. 2006. Group formation in large social networks: membership, growth, and evolution. *KDD.* 44–54.

Lars Backstrom and Jure Leskovec. 2011. Supervised random walks: Predicting and recommending links in social networks. *WSDM*. ACM, 635–644.

Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. 2010. Fast incremental and personalized pagerank. *VLDB* 4, 3 (2010), 173–184.

Albert-Laszlo Barabâsi, Hawoong Jeong, Zoltan Néda, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. 2002. Evolution of the social network of scientific collaborations. *Physica A* 311, 3 (2002), 590–614.

Skye Bender-deMoll and Daniel A. McFarland. 2006. The art and science of dynamic network visualization. *Journal of Social Structure* 7, 2 (2006), 1–38.

Tanya Y Berger-Wolf and Jared Saia. 2006. A framework for analysis of dynamic social networks. *KDD*. ACM, 523–528.

Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. 2009a. Mining graph evolution rules. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 115–130.

Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. 2009b. Mining graph evolution rules. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 115–130.

Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. 2013. CopyCatch: Stopping group attacks by spotting lockstep behavior in social networks. *WWW.* 119–130.

Antje Beyer, Peter Thomason, Xinzhong Li, James Scott, and Jasmin Fisher. 2010. Mechanistic insights into metabolic disturbance during type-2 diabetes and obesity using qualitative networks. In *Transactions on Computational Systems Biology XII*. Springer, 146–162.

Smriti Bhagat, Graham Cormode, and S. Muthukrishnan. 2011. Node classification in social networks. *Social Network Data Analytics*. 115–148.

Ginestra Bianconi and A.-L. Barabási. 2001. Competition and multiscaling in evolving networks. *EPL (Europhysics Letters)* 54, 4 (2001), 436.

Cemal Cagatay Bilgin and Bülent Yener. 2006. Dynamic network evolution: Models, clustering, anomaly detection. *IEEE Networks*.

Petko Bogdanov, Misael Mongiovì, and Ambuj K. Singh. 2011. Mining heavy subgraphs in time-evolving networks. *ICDM*. IEEE, 81–90.

Ulrik Brandes and Steven R Corman. 2003. Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization* 2, 1 (2003), 40–50.

Horst Bunke, Peter J. Dickinson, Miro Kraetzl, and Walter D. Wallis. 2006. *A Graph-Theoretic Approach to Enterprise Network Dynamics*. Vol. 24. Birkhauser, Boston.

Damon Centola, Michael W. Macy, and Victor M. Eguiluz. 2005. Cascade dynamics of multiplex propagation. *arXiv preprint physics/0504165* (2005).

Deepayan Chakrabarti, Christos Faloutsos, and Mary McGlohon. 2010. Graph mining: Laws and generators. *Managing and Mining Graph Data*. 69–123.

Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. 2006. Evolutionary clustering. *KDD*. 554–560.

Jeffrey Chan, James Bailey, and Christopher Leckie. 2008. Discovering correlated spatio-temporal changes in evolving graphs. *Knowledge and Information Systems* 16, 1 (2008), 53–96.

Chaomei Chen. 2006. CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *JASIS* 57, 3 (2006), 359–377.

Chaomei Chen and Steven Morris. 2003. Visualizing evolving networks: Minimum spanning trees versus pathfinder networks. *INFOVIS*. IEEE, 67–74.

Ed H. Chi, James Pitkow, Jock Mackinlay, Peter Pirolli, Rich Gossweiler, and Stuart K. Card. 1998. Visualizing the evolution of web ecologies. *ACM SIGCHI Conference.* 400–407.

Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L. Tseng. 2009. On evolutionary spectral clustering. *TKDD* 3, 4 (2009).

Sutanay Choudhury, Lawrence Holder, George Chin, and John Feo. 2011. Large-scale continuous subgraph queries on streams. *High Performance Computing Meets Databases*. ACM, 29–32.

Atish Das Sarma, Sreenivas Gollapudi, and Rina Panigrahy. 2008. Estimating pagerank on graph streams. *SIGMOD*. ACM, 69–78.

Pedro O. S. Vaz De Melo, Leman Akoglu, Christos Faloutsos, and Antonio A. F. Loureiro. 2010. Surprising patterns for the call duration distribution of mobile phone users. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 354–369.

Prasanna Desikan, Nishith Pathak, Jaideep Srivastava, and Vipin Kumar. 2005. Incremental page rank computation on evolving graphs. *WWW*. 1094–1095.

Patrick Doreian and Frans Stokman. 2013. *Evolution of Social Networks*. Vol. 1. Routledge.

Sergei N. Dorogovtsev and José F. F. Mendes. 2003. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press.

Sergey N. Dorogovtsev, José Fernando F. Mendes, and Alexander N. Samukhin. 2000. Structure of growing networks with preferential linking. *Physical Review Letters* 85, 21 (2000), 4633–4636.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. 2006. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM J. Comput.* 36, 1 (2006), 184–206.

Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM TKDD* 5, 2 (2011), 10.

Ahmed Eldawy, Rohit Khandekar, and Kun-Lung Wu. 2012. Clustering streaming graphs. *ICDCS*. 466–475.

Tanja Falkowski, Jorg Bartelheimer, and Myra Spiliopoulou. 2006. Mining and visualizing the evolution of subgroups in social networks. *International Conference on Web Intelligence*. 52–58.

Tanja Falkowski, Anja Barth, and Myra Spiliopoulou. 2008. Studying community dynamics with an incremental graph mining algorithm. *Conference on Information Systems*. 1–11.

Jure Ferlez, Christos Faloutsos, Jure Leskovec, Dunja Mladenic, and Marko Grobelnik. 2008. Monitoring network evolution using MDL. *ICDE*. 1328–1330.

Johannes Gehrke, Paul Ginsparg, and Jon Kleinberg. 2003. Overview of the 2003 KDD Cup. *SIGKDD Explor. Newsl.* 5, 2 (Dec. 2003), 149–151. DOI:http://dx.doi.org/10.1145/980972.980992

Michaela Goetz, Jure Leskovec, Mary McGlohon, and Christos Faloutsos. 2009. Modeling blog dynamics. *ICWSM*.

Robert Görke, Pascal Maillard, Christian Staudt, and Dorothea Wagner. 2010. *Modularity-Driven Clustering of Dynamic Graphs*. Springer.

Derek Greene, Dónal Doyle, and Pádraig Cunningham. 2010. Tracking the evolution of communities in dynamic social networks. *ASONAM*. IEEE, 176–183.

İsmail Güneş, Zehra Çataltepe, and Şule Gündüz-Öğüdücü. 2013. GA-TVRC-Het: Genetic algorithm enhanced time varying relational classifier for evolving heterogeneous networks. *DMKD.* 1–32.

Ting Guo, Lianhua Chi, and Xingquan Zhu. 2013. Graph hashing and factorization for fast graph stream classification. *CIKM*. 1607–1612.

Manish Gupta, Charu Aggarwal, and Jiawei Han. 2011a. Finding top-k shortest path distance changes in an evolutionary network. *SSTD*. 130–148.

Manish Gupta, Charu Aggarwal, Jiawei Han, and Yizhou Sun. 2011b. Evolutionary clustering and analysis of bibliographic networks. *ASONAM*. 63–70.

Manish Gupta, Jing Gao, Yizhou Sun, and Jiawei Han. 2012a. Community trend outlier detection using soft temporal pattern mining. *ECML/PKDD*. 692–708.

Manish Gupta, Jing Gao, Yizhou Sun, and Jiawei Han. 2012b. Integrating community matching and outlier detection for mining evolutionary community outliers. *KDD*. 859–867.

Mohammad Al Hasan and Mohammed J. Zaki. 2011. A survey of link prediction in social networks. *Social Network Data Analytics*. 243–275.

Xiaofei He, Deng Cai, and Partha Niyogi. 2005. Tensor subspace analysis. *NIPS.* 499–506.

John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. 2004. Tracking evolving communities in large linked networks. *PNAS* 101, Suppl 1 (2004), 5249–5253.

Zan Huang, Xin Li, and Hsinchun Chen. 2005. Link prediction approach to collaborative filtering. *ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, 141–142.

Zan Huang and Dennis K. J. Lin. 2009. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing* 21, 2 (2009), 286–303.

Tsuyoshi Idé and Hisashi Kashima. 2004. Eigenspace-based anomaly detection in computer systems. *KDD.* 440–449.

Piotr Indyk, Nick Koudas, and S. Muthukrishnan. 2000. Identifying representative trends in massive time series data sets using sketches. *VLDB*. 363–372.

Ian Jolliffe. 2005. *Principal Component Analysis*. Wiley Online Library.

David R. Karger. 2000. Minimum cuts in near-linear time. *Journal of the ACM* 47, 1 (2000), 46–76.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. *KDD*. 137–146.

Min-Soo Kim and Jiawei Han. 2009. A particle-and-density based evolutionary clustering method for dynamic networks. *PVLDB* 2, 1 (2009), 622–633.

Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. 1999. The web as a graph: Measurements, models, and methods. *COCOON*. 1–17.

Mladen Kolar, Le Song, Amr Ahmed, and Eric P Xing. 2010. Estimating time-varying networks. *The Annals of Applied Statistics* 4, 1 (2010), 94–123.

Paul Krapivsky, Sidney Redner, and Francois Leyvraz. 2000. Connectivity of growing random networks. *Physical Review Letters* 85, 21 (2000), 4629–4632.

Ravi Kumar, Jasmine Novak, and Andrew Tomkins. 2006. Structure and evolution of online social networks. *KDD*, 611–617.

Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tomkins, and Eli Upfal. 2000. The web as a graph. *PODS*. 1–10.

Jian-Huang Lai, Chang-Dong, Wang, and Philip Yu. 2013. Dynamic community discovery in graph streams. *SDM*, 151–161.

Danh Le-Phuoc, Josiane Xavier Parreira, and Manfred Hauswirth. 2012. Linked stream data processing. In *Reasoning Web: Semantic Technologies for Advanced Query Answering*. Springer, 245–289.

Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. 2007. The dynamics of viral marketing. *ACM TWEB* 1, 1 (2007), 5.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. *KDD*. ACM, 497–506.

Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. 2008. Microscopic evolution of social networks. *KDD*. 462–470.

Jure Leskovec, Deepayan Chakrabarti, Jon M. Kleinberg, and Christos Faloutsos. 2005a. Realistic, mathematically tractable graph generation and evolution, using Kronecker multiplication. *PKDD*. 133–145.

Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. 2005b. Graphs over time: Densification laws, shrinking diameters and possible explanations. *KDD*. 177–187.

Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie S. Glance, and Matthew Hurst. 2007. Patterns of cascading behavior in large blog graphs. *SDM*, 551–556.

Ian XY Leung, Pan Hui, Pietro Lio, and Jon Crowcroft. 2009. Towards real-time community detection in large networks. *Physical Review E* 79, 6 (2009), 066107.

Michael Ley. 2002. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *String Processing and Information Retrieval*. Springer, 1–10.

Bin Li, Xingquan Zhu, Lianhua Chi, and Chengqi Zhang. 2012. Nested subtree hash kernels for large-scale graph classification over streams. *ICDM*. 399–408.

David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *JASIS* 58, 7 (2007), 1019–1031.

Cindy Xide Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. 2010. PET: A statistical model for popular events tracking in social communities. *KDD*. ACM, 929–938.

Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. 2008. Facetnet: A framework for analyzing communities and their evolutions in dynamic networks. *WWW*. 685–694.

Siyuan Liu, Lei Li, Christos Faloutsos, and Lionel M. Ni. 2011. Mobile phone graph evolution: Findings, model and interpretation. *ICDM Workshops*. 323–330.

Fragkiskos D. Malliaros, Vasileios Megalooikonomou, and Christos Faloutsos. 2012. Fast robustness estimation in large social graphs: Communities and anomaly detection. *SDM*, 942–953.

Mary McGlohon, Leman Akoglu, and Christos Faloutsos. 2011. Statistical properties of social networks. *Social Network Data Analytics*. 17–42.

Mary McGlohon, Jure Leskovec, Christos Faloutsos, Matthew Hurst, and Natalie S. Glance. 2007. Finding patterns in blog shapes and blog evolution. *ICWSM*.

Andrew McGregor. 2005. Finding graph matchings in data streams. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. Springer, 170–181.

Misael Mongiovi, Petko Bogdanov, Razvan Ranca, Evangelos E Papalexakis, Christos Faloutsos, and Ambuj K Singh. 2013. NetSpot: Spotting significant anomalous regions on dynamic networks. *SDM*. SIAM.

James Moody, Daniel McFarland, and Skye Bender-deMoll. 2005. Dynamic network visualization1. *American Journal of Sociology* 110, 4 (2005), 1206–1241.

Huazhong Ning, Wei Xu, Yun Chi, Yihong Gong, and Thomas S. Huang. 2007. Incremental spectral clustering with application to monitoring of evolving blog communities. *SDM*, 261–272.

Gergely Palla, Albert Barabási, and Tamás Vicsek. 2007. Quantifying social group evolution. *Nature* 446, 7136 (2007), 664–667.

Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435, 7043 (2005), 814–818.

Shirui Pan and Xingquan Zhu. 2013. Graph classification with imbalanced class distributions and noise. *AAAI*. AAAI Press, 1586–1592.

Shirui Pan, Xingquan Zhu, Chengqi Zhang, and Philip Yu. 2013. Graph stream classification using labeled and unlabeled graphs. *ICDE*. 398–409.

Panagiotis Papadimitriou, Ali Dasdan, and Hector Garcia-Molina. 2010. Web graph similarity for anomaly detection. *Journal of Internet Services and Applications* 1, 1 (2010), 19–30.

B. Pincombe. 2005. Anomaly detection in time series of graphs using ARMA processes. *ASOR Bulletin* 24, 4 (2005), 2.

Alexandrin Popescul and Lyle Ungar. 2003. Statistical relational learning for link prediction. *IJCAI Workshop on Learning Statistical Models from Relational Data*, Vol. 2003. Citeseer.

B. Aditya Prakash and Christos Faloutsos. 2012. Understanding and managing cascades on large graphs. *PVLDB* 5, 12 (2012), 2024–2025.

Carey E. Priebe, John M. Conroy, David J. Marchette, and Youngser Park. 2005. Scan statistics on enron graphs. *Computational & Mathematical Organization Theory* 11, 3 (2005), 229–247.

Usha Nandini Raghavan, Reka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76, 3 (2007), 036106.

Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. *KDD*. ACM, 61–70.

Ryan Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. 2012. Role-dynamics: Fast mining of large dynamic networks. *WWW*. ACM, 997–1006.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: Real-time event detection by social sensors. *WWW*. ACM, 851–860.

Arnaud Sallaberry, Chris Muelder, and Kwan-Liu Ma. 2013. Clustering, visualizing, and navigating for large dynamic graphs. In *Graph Drawing*. Springer, 487–498.

Purnamrita Sarkar, Deepayan Chakrabarti, and Michael I. Jordan. 2012. Nonparametric link prediction in dynamic networks. *ICML*.

Ramesh Sarukkai. 2000. Link prediction and path analysis using Markov chains. *Computer Networks* 33, 1 (2000), 377–386.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2002. Incremental singular value decomposition algorithms for highly scalable recommender systems. *International Conference on Computer and Information Science*. 27–28.

Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. 2009. Event detection and tracking in social streams. *ICWSM*.

D. Seung and L. Lee. 2001. Algorithms for non-negative matrix factorization. *NIPS* 13 (2001), 556–562.

Peter Shoubridge, Miro Kraetzl, Walter Wallis, and Horst Bunke. 2002. Detection of abnormal change in a time series of graphs. *Journal of Interconnection Networks* 3, 01n02 (2002), 85–101.

Peter Shoubridge, Miro Kraetzl, and David Ray. 1999. Detection of abnormal change in dynamic networks. *IDC*. IEEE, 557–562.

J. Silva and R. Willett. 2008. Detection of anomalous meetings in a social network. *Information Sciences and Systems*. IEEE, 636–641.

Tom A. B. Snijders, Christian E. G. Steglich, and Michael Schweinberger. 2007. Modeling the co-evolution of networks and behavior. *Longitudinal Models in the Behavioral and Related Sciences* (2007), 41–71.

Ricard V. Solé, Romualdo Pastor-Satorras, Eric Smith, and Thomas B. Kepler. 2002. A model of large-scale proteome evolution. *Advances in Complex Systems* 5, 1 (2002), 43–54.

Myra Spiliopoulou. 2011. Evolution in social networks: A survey. In *Social Network Data Analytics*. Springer, 149–175.

Isabelle Stanton and Gabriel Kliot. 2012. Streaming graph partitioning for large distributed graphs. *KDD*. 1222–1230.

Joshua M. Stuart, Eran Segal, Daphne Koller, and Stuart K. Kim. 2003. A gene-coexpression network for global discovery of conserved genetic modules. *Science* 302, 5643 (2003), 249–255.

Karthik Subbian, Charu Aggarwal, and Jaideep Srivastava. 2013. Content-centric flow mining for influence analysis in social streams. *CIKM*. ACM, 841–846.

Karthik Subbian and Prem Melville. 2011. Supervised rank aggregation for predicting influencers in Twitter. *Socialcom*. IEEE, 661–665.

Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip Yu. 2007. GraphScope: Parameter-free mining of large time-evolving graphs. *KDD*. 687–696.

Jimeng Sun and Jie Tang. 2011. A survey of models and algorithms for social influence analysis. *Social Network Data Analytics*. 177–214.

Jimeng Sun, Dacheng Tao, and Christos Faloutsos. 2006. Beyond streams and graphs: Dynamic tensor analysis. *KDD*. ACM, 374–383.

Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip Yu, and Christos Faloutsos. 2008. Incremental tensor analysis: Theory and applications. *ACM TKDD* 2, 3 (2008), 11.

Jimeng Sun, Yinglian Xie, Hui Zhang, and Christos Faloutsos. 2007. Less is more: Compact matrix decomposition for large sparse graphs. (2007), *SDM*, 366–377.

Yizhou Sun, Rick Barber, Manish Gupta, Charu C. Aggarwal, and Jiawei Han. 2011. Co-author relationship prediction in heterogeneous bibliographic networks. *ASONAM*. IEEE, 121–128.

Yizhou Sun, Jiawei Han, Charu Aggarwal, and Nitesh V. Chawla. 2012. When will it happen?: Relationship prediction in heterogeneous information networks. *WSDM*. 663–672.

Yizhou Sun, Jie Tang, Jiawei Han, Manish Gupta, and Bo Zhao. 2010. Community evolution detection in dynamic heterogeneous information networks. *Mining and Learning with Graphs (MLG)*. ACM, 137–146.

Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based clustering of heterogeneous information networks with star network schema. *KDD*. 797–806.

Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaıane. 2013. Incremental local community identification in dynamic social networks. *ASONAM* (2013), 90–94.

Lei Tang, Huan Liu, Jianping Zhang, and Zohreh Nazeri. 2008. Community evolution in dynamic multi-mode networks. *KDD*. ACM, 677–685.

Chayant Tantipathananandh, Tanya Berger-Wolf, and David Kempe. 2007. A framework for community identification in dynamic social networks. *KDD*. ACM, 717–726.

Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. 2003. Link prediction in relational data. *NIPS*, Vol. 15.

Sarah A. Teichmann and M. Madan Babu. 2004. Gene regulatory network growth by duplication. *Nature Genetics* 36, 5 (2004), 492–496.

Hanghang Tong, Spiros Papadimitriou, Jimeng Sun, Philip Yu, and Christos Faloutsos. 2008a. Colibri: Fast mining of large static and dynamic graphs. *KDD*. 686–694.

Hanghang Tong, Spiros Papadimitriou, Philip Yu, and Christos Faloutsos. 2008b. Fast monitoring proximity and centrality on time-evolving bipartite graphs. *Statistical Analysis and Data Mining* 1, 3 (2008), 142–156.

Hanghang Tong, Yasushi Sakurai, Tina Eliassi-Rad, and Christos Faloutsos. 2008c. Fast mining of complex time-stamped events. *CIKM*. ACM, 759–768.

Koji Tsuda and Hiroto Saigo. 2010. Graph classification. *Managing and Mining Graph Data*. 337–363.

Tomasz Tylenda, Ralitsa Angelova, and Srikanta Bedathur. 2009. Towards time-aware link prediction in evolving social networks. *Social Network Mining and Analysis Workshop*. 9.

Alexei Vázquez, Alessandro Flammini, Amos Maritan, and Alessandro Vespignani. 2002. Modeling of protein interaction networks. *Complexus* 1, 1 (2002), 38–44.

Jeffrey Scott Vitter. 1985. Random sampling with a reservoir. *ACM Transactions in Mathematical Software* 11, 1 (1985), 37–57.

Changliang Wang and Lei Chen. 2009. Continuous subgraph pattern search over graph streams. *ICDE*. 393–404.

Lijun Wang, Manjeet Rege, Ming Dong, and Yongsheng Ding. 2012. Low-rank kernel matrix factorization for large-scale evolutionary clustering. *TKDE* 24, 6 (2012), 1036–1050.

Kevin S. Xu, Mark Kliger, and A. O. Hero. 2010. Evolutionary spectral clustering with adaptive forgetting factor. *Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2174–2177.

Tianbing Xu, Zhongfei Zhang, Philip S. Yu, and Bo Long. 2012. Generative models for evolutionary clustering. *ACM TKDD* 6, 2 (2012), 7.

Leiming Yan, Jinwei Wang, Jin Han, and Yuxiang Wang. 2012. A significance-driven framework for characterizing and finding evolving patterns of news networks. *Artificial Intelligence and Computer Intelligence*. Springer, 134–141.

Weiren Yu, Charu C. Aggarwal, Shuai Ma, and Haixun Wang. 2013. On anomalous hotspot discovery in graph streams. *ICDM*, 1271–1276.

Jian Zhang. 2010. A survey on streaming algorithms for massive graphs. In *Managing and Mining Graph Data*. Springer, 393–420.

Peixiang Zhao, Charu C. Aggarwal, and Min Wang. 2011. Gsketch: On query estimation in graph streams. *VLDB* 5, 3 (2011), 193–204.

Qiankun Zhao, Prasenjit Mitra, and Bi Chen. 2007. Temporal and information flow based event detection from social text streams. *NCAI*, Vol. 22, 1501.

Yuchen Zhao and Philip Yu. 2013. On graph stream clustering with side information. *SDM*, 139–150.

Elena Zheleva, Hossam Sharara, and Lise Getoor. 2009. Co-evolution of social and affiliation networks. *KDD*. ACM, 1007–1016.

Received July 2013; revised April 2014; accepted April 2014