LLNL-CONF-603132

Modeling Dynamic Behavior in Large Evolving Graphs

R. A. Rossi, B. Gallagher, J. Neville, K. Henderson

November 19, 2012

ACM International Conference on Web Search and Data Mining
Rome, Italy
February 4, 2013 through February 8, 2013

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Modeling Dynamic Behavior in Large Evolving Graphs

Ryan A. Rossi
Jennifer Neville
Purdue University
{rrossi, neville}@purdue.edu

Brian Gallagher
Keith Henderson
Lawrence Livermore National Laboratory
{bgallagher, keith}@llnl.gov

ABSTRACT

Given a large time-evolving graph, how can we model and characterize the temporal behaviors of individual nodes (and network states)? How can we model the behavioral transition patterns of nodes? We propose a temporal behavior model that captures the "roles" of nodes in the graph and how they evolve over time. The proposed *dynamic behavioral mixed-membership model* (DBMM) is scalable, fully automatic (no user-defined parameters), non-parametric/data-driven (no specific functional form or parameterization), interpretable (identifies explainable patterns), and flexible (applicable to dynamic and streaming networks). Moreover, the interpretable behavioral roles are generalizable and computationally efficient. We applied our model for (a) identifying patterns and trends of nodes and network states based on the temporal behavior, (b) predicting future structural changes, and (c) detecting unusual temporal behavior transitions. The experiments demonstrate the scalability, flexibility, and effectiveness of our model for identifying interesting patterns, detecting unusual structural transitions, and predicting the future structural changes of the network and individual nodes.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Experimentation

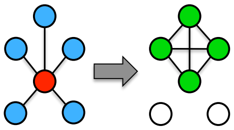
Keywords

Graph mining, dynamic network models, dynamic roles

1. INTRODUCTION

In recent years, we have witnessed a tremendous growth in both the variety and scope of network datasets. In particular, network datasets often record the interactions and/or transactions among a set of entities—for example, personal communication (e.g., email, phone), online social network interactions (e.g., Twitter, Facebook), web traffic between servers and hosts, and router traffic among autonomous systems. A notable characteristic of these *activity* networks, is that the structure of the networks change over time (e.g., as people communicate with different friends). These temporal dynamics are key to understanding system *behavior*, thus it is critical to model and predict the network changes over time. An improved understanding of temporal patterns will facilitate for example, the development of software systems to optimally manage data flow, to detect fraud and intrusions, and to allocate resources for growth over time.

Although some recent research has focused on the analysis of dynamic networks [17, 3, 5, 12, 4, 21], there has been less work on developing *models* of temporal behavior in large scale network datasets. There has been some work on modeling temporal events in large scale networks [2, 31] and other work that uses temporal link and attribute patterns to improve predictive models [26]. In addition, there is work on identifying clusters in dynamic data [5, 27] but these methods focus on discovering underlying communities over time—sets of nodes that are densely connected together. In contrast, we are interested in uncovering the *behavioral* patterns of nodes in the graph and modeling how those patterns change over time. In the simple example below, the red node is a "star-center" and the blue represents peripheral nodes. At some later time, the top nodes transition to a "near-clique" while the two bottom nodes become inactive.



The recent work on dynamic mixed-membership stochastic block models (dMMSB: [9, 30]), is to our knowledge, one of the only methods suitable for modeling node-centric properties over time. The dMMSB model identifies groups of nodes with similar patterns of linkage and characterizes how group memberships change over time. However, dMMSB assumes a specific parametric form where the groups are defined through linkage to specific nodes (i.e., in particular types of groups) rather than more general forms of node behavior over dynamic node sets. More importantly the dMMSB estimation algorithm is not scalable, which makes the method unsuitable for analysis of large graphs.

In this work, we aim to develop a descriptive model to answer the following questions for dynamic network datasets:

1. **Identify dynamic patterns in node behavior.** What types of high level temporal patterns and trends do the data exhibit? Are behaviors cyclical or predictable? Do nodes have different behavioral patterns?
2. **Predict future structural changes.** Can we predict

WSDM'13, February 4–8, 2013, Rome, Italy.
Copyright 2013 ACM 978-1-4503-1869-3/13/02 ...$10.00.

when a node's role will change (e.g., a node with high in-degree transitions to a node with high betweenness)? Is the overall structure of the graph becoming more or less predictable over time?

3. **Detect unusual transitions in behavior.** Are there nodes and/or points in time with significantly different behavioral patterns?

To facilitate the investigation of these questions, we propose to model node "roles" and how they change over time. Informally, *roles* can be viewed as sets of nodes that are more structurally similar to nodes inside the set than outside whereas communities are sets of nodes with more connections inside the set than outside. Specifically, to focus on *node behavior* (rather than the complimentary concept of community finding) we use non-parametric feature-based roles.

Using these non-parametric roles, which can generalize to new unseen nodes, we propose a novel *dynamic behavioral mixed-membership model* (DBMM) suitable for large, unbounded, time-evolving networks. The DBMM discovers features (i.e., using the graph and intrinsic attributes), extracts these features for all timesteps, and automatically learns behavioral "roles" for nodes at each timestep. The number of behavioral roles are selected automatically using MDL or AIC. Afterwards, we learn behavior transition matrices for each node (i.e., given a node role r_i, what is the probability of transitioning to r_j at the next point in time).

Our proposed DBMM technique allows us to investigate the properties of temporal networks and understand both global and local behaviors, detect anomalies, as well as predict future structural changes. The main strengths of the approach include:

- ⋆ **Automatic.** The algorithm doesn't require user-defined parameters.
- ⋆ **Scalable.** The learning algorithm is linear in the number of edges in the time-interval under consideration. It is also easily parallelizable as features, roles, transition models can be learned independently at each time.
- ⋆ **Non-parametric and data-driven.** The model structure (i.e., number of parameters) and more generally the parameterization depends on the properties of the time-evolving network.
- ⋆ **Interpretable and intuitive.** The DBMM is based on an intuitive behavioral representation (structural properties) of the network and individual nodes. It identifies explainable patterns, trends, and aids in understanding the underlying dynamic process.
- ⋆ **Flexible.** The definition of behavior in our model can be tuned for specific applications. The algorithm is applicable for all types of time-evolving networks.

We demonstrate the application our our model on several real world datasets, showing that it both accurately predicts future structural changes as well as identifying interesting temporal patterns and anomalies. We discuss the scalability of the approach and notably we apply the DBMM to networks with up to 300,000 nodes and 4 million edges—datasets that are orders of magnitude larger than could be modeled with dMMSBs.

2. DYNAMIC BEHAVIORAL MODEL

Our goal is to model the behavioral roles of nodes and their evolution over time. Given a sequence of network snapshots (graphs and attributes), the Dynamic Behavioral Mixed Membership Model (DBMM) consists of (1) automatically learning a set of representative features, (2) extracting features from each graph, (3) discovering behavioral roles (4) iteratively extracting these roles from the sequence of network snapshots over time and (5) learning a predictive model of how these behaviors change over time. As an aside, let us note that DBMM is a *scalable general framework* for analyzing temporal behavior as the model components can be replaced by others and each component can be appropriately tuned for any application (e.g., for the feature set, any feature construction system from [25] can conceivably be used).

2.1 Data Model for Temporal Networks

Given a dynamic network $\mathcal{D} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges in $\mathcal{D}$, a network snapshot $\mathcal{S}_t = (\mathcal{N}_t, \mathcal{E}_t)$ is a subgraph of $\mathcal{D}$ where $\mathcal{E}_t$ are the edges in $\mathcal{E}$ active at time t and $\mathcal{N}_t$ are the endpoints of the edges $\mathcal{E}_t$.

2.2 Representing Network Behavior

The idea is to discover a set of underlying roles, which together describe the behaviors observed in the network, and then assign a probability distribution over these roles to each node in the network, which explain that node's observed behavior. Roles are extracted via a two-step process.

Feature Discovery. The first step is to represent each active node in a given snapshot graph $\mathcal{S}_t$ using a set of representative features. For this task, we leverage [14]. The method constructs degree and egonet measures (in / out, weighted, ...), then aggregates these measures using sum (or mean) creating recursive features. After each aggregation step, correlated features are pruned using logarithmic binning. The aggregation proceeds recursively, until there are no new features. Formally, we discover a set of features at time t denoted $\boldsymbol{V}_t$ such that $\boldsymbol{V}_t$ is an $n_t \times f$ matrix where n_t is the number of active nodes and f is the number of features learned from the snapshot graph $\mathcal{S}_t$. The features are extracted for each network snapshot resulting in a sequence of node-feature matrices, denoted $\boldsymbol{V} = \{\boldsymbol{V}_t : t = 1, ..., t_{max}\}$.

Role Discovery. The next step is to automatically discover groups of nodes (representing common patterns of behavior) based on their features. For this purpose, we use Non-negative Matrix Factorization (NMF) to extract roles [15] and extend it for a sequence of graphs. Given a sequence of node-feature matrices, we generate a rank-r approximation $\boldsymbol{G}_t\boldsymbol{F} \approx \boldsymbol{V}_t$ where each row of $\boldsymbol{G}_t \in \mathbb{R}^{n \times r}$ represents a node's membership in each role and each column of $\boldsymbol{F} \in \mathbb{R}^{r \times f}$ represents how membership of a specific role contributes to estimated feature values. For constructing the "closest" rank-r approximation we use NMF (multiplicative update method) because of interpretability and efficiency, though any other method for constructing such an approximation may be used instead (SVD, spectral decomposition). More formally, given a non-negative matrix $\mathbf{V}_t \in \mathbb{R}^{n_t \times f}$ and a positive integer $r < \min(n_t, f)$, find non-negative matrices $\mathbf{G}_t \in \mathbb{R}^{n_t \times r}$ and $\mathbf{F} \in \mathbb{R}^{r \times f}$ that minimizes the functional,

$$f(\mathbf{G}_t, \mathbf{F}) = \frac{1}{2}||\mathbf{V}_t - \mathbf{G}_t\mathbf{F}||_F^2$$

We iteratively estimate the node-role memberships for each network snapshot $\boldsymbol{G} = \{\boldsymbol{G}_t : t = 1, ..., t_{max}\}$ given $\boldsymbol{F}$ and $\boldsymbol{V} = \{\boldsymbol{V}_t : t = 1, ..., t_{max}\}$ using NMF. Afterwards, we

Table 1: Dataset characteristics. The number of learned features and roles provide intuition about the underlying generative process and also indicates the amount of complexity present in the network.

Dataset	Features	Roles	Nodes	Edges	|T|	length
Twitter Relationships	1325	12	310,809	4,095,627	41	1 day
Twitter (Copenhagen)	150	5	8,581	27,889	112	3 hours
Facebook	161	9	46,952	183,831	18	1 day
Email-Univ	652	10	116,893	1,270,285	50	60 min
Network-Trace	268	11	183,389	1,631,824	49	15 min
Internet AS	30	2	37,632	505,772	28	3 months
Enron	173	6	151	50,572	82	2 weeks
IMDB	45	3	21,257	296,188	28	1 year
Reality	99	5	97	31,694	46	1 month

have a sequence of matrices $\boldsymbol{G}_1, \boldsymbol{G}_2, ..., \boldsymbol{G}_t, ..., \boldsymbol{G}_{t_{max}}$ where each active node at time t is represented with their current role memberships.

The number of structural roles r is automatically selected using Minimum Description Length (MDL) criterion. However, AIC or any model selection may be used instead. Intuitively, learning more roles, increases model complexity, but decreases the amount of errors. Conversely, learning less roles, decreases model complexity, but increases the amount of errors. In this way, MDL selects the number of behavioral roles r such that the model complexity (number of bits) and model errors are balanced. Naturally, the best model minimizes, $number\ of\ bits + errors$. See [15] for more details.

2.3 Behavioral Transition Model

Given a sequence of dynamic behaviors $\boldsymbol{G} = \{\boldsymbol{G}_t : t = 1, ..., t_{max}\}$, we can learn a model of how behavior in our network changes over time. More formally, given two behavioral snapshots, $\boldsymbol{G}_{t-1}$ and $\boldsymbol{G}_t$, we learn a transition matrix $\boldsymbol{T} \in \mathbb{R}^{r \times r}$ that approximates the change in behavior from time $t-1$ to t. The transition matrix $\boldsymbol{T}$ represents how likely a node is to transition from role r_i to role r_j for that particular time interval:

$$\boldsymbol{T} = \begin{bmatrix} z^{(r_1 \to r_1)} & z^{(r_1 \to r_2)} & \dots & z^{(r_1 \to r_m)} \\ z^{(r_2 \to r_1)} & z^{(r_2 \to r_2)} & \dots & z^{(r_2 \to r_m)} \\ \vdots & \dots & \ddots & \dots \\ z^{(r_m \to r_1)} & z^{(r_m \to r_2)} & \dots & z^{(r_m \to r_m)} \end{bmatrix}$$

where $\boldsymbol{T}$ is estimated using NMF such that $\boldsymbol{G}_{t-1}\boldsymbol{T} \approx \boldsymbol{G}_t$.

In the simple form of the model presented above, we learn $\boldsymbol{T}$ using only a single transition (i.e., $t-1$ to t). However, we also propose variations that leverage more available data by considering multiple transitions (*stacked model*) or that smooth over a sequence of transitions using kernel functions (*summary model*). We discuss these in detail next.

2.3.1 Stacked Transition Model

The stacked model uses the training examples from the k previous timesteps. More formally, the stacked model is defined as,

$$\begin{bmatrix} \boldsymbol{G}_{t-1} \\ \boldsymbol{G}_{t-2} \\ \vdots \\ \boldsymbol{G}_{k-1} \end{bmatrix} \boldsymbol{T} \approx \begin{bmatrix} \boldsymbol{G}_{t} \\ \boldsymbol{G}_{t-1} \\ \vdots \\ \boldsymbol{G}_{k} \end{bmatrix}$$

where $k = t - w$ and w is the window size; typically $w = 10$. Let us denote the stacked behavioral snapshots as $\boldsymbol{G}_{k:t}$ where $k : t$ represents all the training examples from timestep k to timestep t.

2.3.2 Summary Transition Model

This class of models uses k previous timesteps to weight the training examples at time t using some kernel function. The exponential decay and linear kernels are used in this work. The temporal weights can be viewed as probabilities that a node behavior is still active at the current time step t, given that it was observed at time $(t-k)$. We define the summary behavioral snapshot $\boldsymbol{G}_{S(t)}$ as a weighted sum of the temporal role-memberships up to time t as follows, $\boldsymbol{G}_{S(t)} = \alpha_1\boldsymbol{G}_k + ... + \alpha_{w-1}\boldsymbol{G}_{t-1} + \alpha_w\boldsymbol{G}_t = \sum_{i=k}^{t} K(\boldsymbol{G}_i; t, \theta)$ where α determines the contribution of each snapshot in the summary model.

In addition to exponential and linear kernels, we experimented with the inverse linear and also tried various θ values. Overall, we found the linear kernel (and exponential) to be the most accurate with $\theta = 0.7$. Nevertheless, the optimal θ will depend on the type of dynamic network and the volatility.

2.4 Remarks

For each type of transition model (e.g., stacked or summary), we may learn a *global transition model* that describes how the behavior of the network as a whole changes over time or we may learn a *local transition model* for each individual node. The local transition model describes more precisely how the behavioral roles of that individual node change over time. We can estimate the local transition model for a node i as $\boldsymbol{G}_{t-1}^{(i)}\boldsymbol{T}^{(i)} \approx \boldsymbol{G}_t^{(i)}$ using NMF. The global transition model for the network is estimated in exactly the same way as described above in §2.3.

We have found the summary model to be the best performer for prediction tasks because of its ability to smooth over multiple timesteps. However, for precisely this reason, the summary model is more difficult to interpret. Therefore, we use the summary model for prediction tasks and the stacked representation for data analysis tasks, due to its interpretability. Let us note that to achieve better accuracy in predictions, one may also estimate local transition models for each node and use these for predicting a node's future role memberships. All of these options make our model flexible for use in a variety of applications.

We also experimented with other variants of the DBMM transition model, including a stacked-summary hybrid and multi-state models, which make an explicit distinction between transitions from activate states and transitions from inactive states. However, we opted in favor of the simpler

Features	S-Center	S-Edge	Bridge	Clique
S-Center	**0.08**	0.25	0.34	0.33
S-Edge	0.27	**0.11**	0.25	0.37
Bridge	0.29	0.20	**0.17**	0.34
Clique	0.24	0.24	0.29	**0.23**

Roles	S-Center	S-Edge	Bridge	Clique
S-Center	**0.07**	0.25	0.33	0.35
S-Edge	0.28	**0.10**	0.22	0.40
Bridge	0.29	0.18	**0.16**	0.37
Clique	0.24	0.25	0.29	**0.22**

Table 2: Validating DBMM's ability to distinguish patterns. Note C is row-normalized.

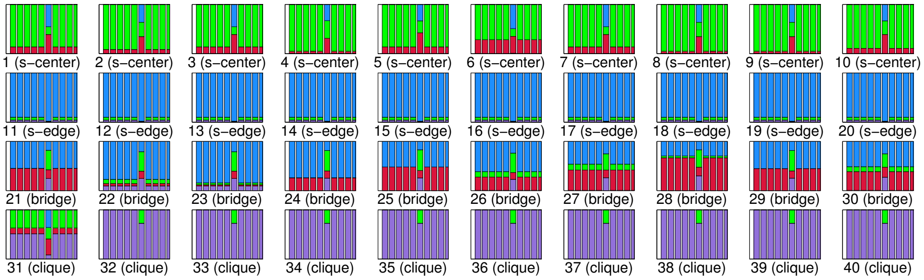


Figure 1: The pattern of each node is listed below the mixed-membership plot whereas the colors represent roles learned from our model. For simplicity, the node's pattern-type is kept stable over time. Strikingly, the DBMM clearly reveals the underlying patterns of the nodes as each pattern has a distinct signature in terms of the role distribution. For instance, the blue role of a bridge node indicates the local similarity with that of a star-edge node (low degree,...) while the red role captures the bridges more global and intrinsic property of acting as a backbone for the other nodes. The other patterns are even more straightforward to interpret. We also inject a type of global anomaly at $t = 6$ (bridges connecting to each other) which is clearly revealed as such in the plots.

stacked and summary models because none of these other models provided an obvious advantage.

While our model currently assumes the role definitions are somewhat stationary, we have found that these roles generalize and can even be applied across different networks. Nevertheless, to remove this assumption, we could simply track the loss over time and recompute the roles when it surpasses some threshold.

3. EXPLORATORY ANALYSIS

This section explores the effectiveness of DBMM for dynamic network analysis tasks.

3.1 Datasets & Structural Analysis

We apply the DBMM model using a variety of dynamic networks from different domains [23]. See Table 1 for details. Interestingly, we find a relationship between the complexity of DBMM and the complexity present in the graph. This is clearly clearly shown in Table 1 by analyzing simple measures generated from the DBMM behavioral representation such as the number of learned features and the number of roles. For instance, the Internet AS topology has some hierarchical structure or recurring patterns of connectivity among ISPs and therefore our model discovers only 30 features. This is in contrast to networks with more complex patterns of connectivity such as twitter and other transaction networks like the email network. In these cases, the links are instantaneous and might only last for some duration of time, thus making more complex structures more likely.

3.2 Experiments on Synthetic Data

In this section, we demonstrate the ability of DBMM to distinguish between common graph patterns (and consequently recover the synthetic roles). For this task, we design a simple graph generator that constructs graphs probabilistically with four main patterns: 'center of a star', 'edge of a star', bridge nodes (connecting stars/cliques), and clique nodes. After constructing the graph, we validate that the DBMM model captured these patterns by measuring if the extracted features and roles represent the known probabilistic patterns. We do this by computing the pairwise euclidean distance matrix $\boldsymbol{D}$ using the initial feature matrix $\boldsymbol{V}$ (and role-membership matrix $\boldsymbol{G}$). Let r_i denote the actual pattern of node i, and $\mathcal{P} = \{(i,j) | r_i = p,\ r_j = q\}$ then $C_{p,q} = \sum_{(i,j)\in\mathcal{P}} D_{i,j}$.

Clearly, the roles and features from nodes of the same pattern are shown to be more similar than the others (smaller values along the diagonal). See Table 2.3. Additionally, the patterns that are structurally similar to one another are represented as such by our model (star-center and clique). In Fig. 1, we visualize the mixed-memberships of 10 randomly chosen nodes from each pattern-type. Each pattern has a distinct and consistent signature in terms of the role distribution.

3.3 Interpretation & Analysis

We start with an illustrative example of applying DBMM to a large IP trace network, shown in Figure 2. We first plot the time-evolving mixed-memberships from four nodes shown in Figure 2(b) and then visualize their corresponding transition models in Figure 2(a). In the time-evolving mixed-memberships, inactivity is represented by white bars whereas in the transition models inactivity corresponds to the last row/column. The transition models are learned using the stacked representation which aids in the understandability and interpretation of the roles and their modeled transitions.

The time-evolving mixed-memberships for each of the four example nodes in Figure 2(b) show distinct patterns from one another which are easy to identify. The four patterns represented by these nodes can be classified as having the following patterns of structural behavior,

1. **Structural Stability.** This node's structural behavior (and communication pattern) doesn't is relatively stable over the time.
2. **Homogeneous.** The node for the most part takes on a single behavioral role.

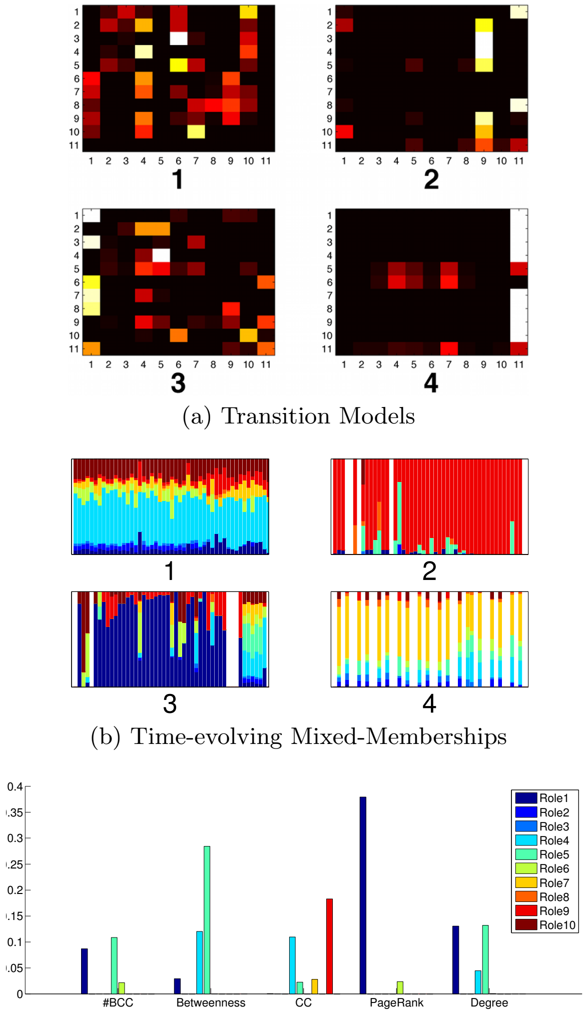
3. **Abrupt transition.** Their structural behavior changes abruptly. In the IP-network, it could be that the IP was released and someone was assigned it or perhaps that the machine was compromised and began acting maliciously.
4. **Periodic activity.** The node has periodic activity, but maintains similar structural behavior. In the case of the IP-communication network, this machine could be infected and every 30 minutes sends out a communication to the master indicating it's connected and "listening".

For the four example nodes, we show their transition models in Figure 2(a). The transition models represent the probability of transitioning or taking on the structural behavior of role j given that your current role (or main role) is role i. For instance, node 2 homogeneously takes on the red role over time as discussed previously. From Figure 2(c), we see that the red role is "role 9", and looking back at the node's learned transition model, we find that column 9 contains most of the mass, which represents that their is a high probability of transitioning from any other role to the red role. As shown in the mixed-memberships over time, this is exactly what is expected. As another example, we find that node 4 usually transitions from a mix of active roles to the inactive role (i.e., the inactive role is represented by column/row eleven). Therefore, we would expect our learned transition model to capture this by placing most of the mass on the last column, representing the probability of going inactive after having a mix of active roles in the previous timestep, which is exactly what we see in the fourth transition model.

Instead of providing subjective or anecdotal evidence for what the roles represent, we interpret the roles of the DBMM with respect to well-known node measurements (e.g., degree, clustering coefficient, betweenness,...). We extend the analytical tools from [15] for use in interpreting the role dynamics. The first technique interprets the roles using the dynamic node-role memberships $\boldsymbol{G}_t$ and a node measure matrix $\boldsymbol{M}_t \in \mathbb{R}^{n \times m}$ to compute a non-negative matrix $\boldsymbol{E}_t$ such that $\boldsymbol{G}_t \boldsymbol{E}_t \approx \boldsymbol{M}_t$. The node measurements used are betweenness, biconnected components, PageRank, clustering coefficient, and degree. The matrix $\boldsymbol{E}_t$ represents the contributions of the node measures to the roles at time t. We report average contributions over time.

Figure 2(c) shows this quantitative interpretation of roles for the IP network. Intuitively, the first role represents nodes with high PageRank, while role five represents nodes with high betweenness, whereas role nine represents nodes with large clustering coefficient. The other roles represent more specialized structural motifs that were not captured by the set of traditional measures used for interpretation.

The DBMM can be used to understand the temporal behavior across a variety of time-evolving networks. Figure 3 shows another example for an email communication network. Just as before, we can identify significant trends and patterns and interpret these using the role interpretations from Figure 3(b). One notable behavioral pattern in the email communications is that most users have a set of roles for the daytime and a different set for night. Intuitively, one set of roles is work related and the other is more personal/family related (see nodes 1 & 2, among others). We also find nodes that have inconsistent or unstable behavior over the time, such as 17, 18, 19, among others. Additionally, some nodes have relatively stable structural behavior over the two days, such as node 4. This is also unusual, since one might expect a user's behavior to change from the work hours to the evening/night. However, users that are consistently dominated by multiple active roles are of importance (may serve in managerial or leadership roles), since they connect to groups of nodes with different types of structural patterns (see nodes 5-7).



(a) Transition Models

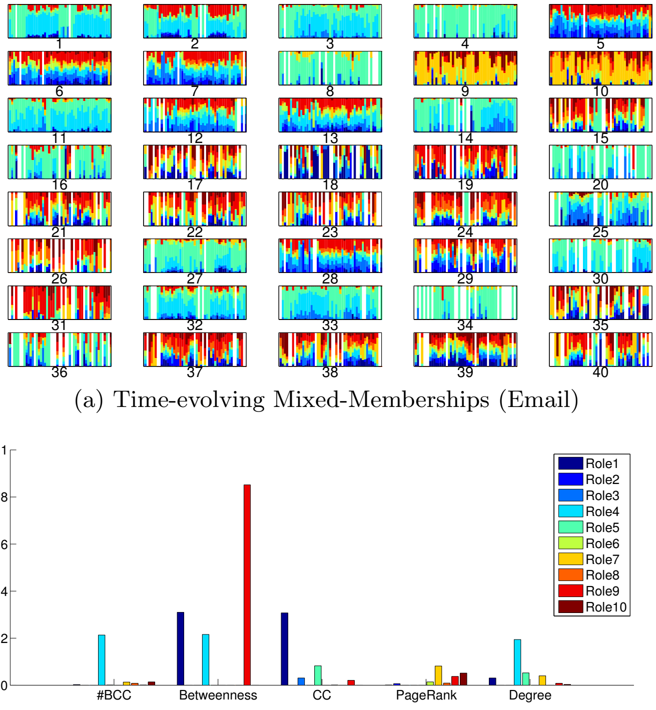
(b) Time-evolving Mixed-Memberships

(c) Role Interpretation

Figure 2: The DBMM transition model effectively captures the diverse temporal behavior of hosts in a computer network. (a) Transition matrices for 4 hosts. The y-axis represents the role the node transitions from, the x-axis is the role we transition to. Inactivity is represented by the last row/column. (b) Corresponding role-memberships over time. The x-axis represents time while the y-axis represents the role distribution at each point in time. Each distinct color represents a learned role. (c) Characteristics of individual roles.

3.4 Clustering Temporal Behaviors

Lastly to show the patterns of the learned transition matrices, we cluster nodes based on their temporal behaviors. We find that this clustering reveals the underlying structural patterns of the evolving mixed-memberships. Formally, let $\boldsymbol{T}^{(i)}$ and $\boldsymbol{T}^{(j)}$ be the transition matrices of two nodes i and j. Then we create an $r \times r$ vector from each of the node



(a) Time-evolving Mixed-Memberships (Email)

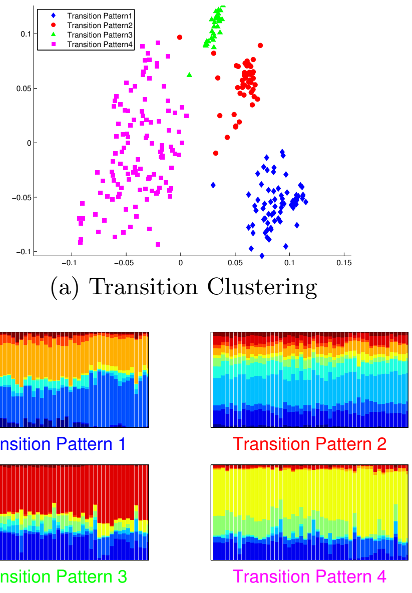
(b) Email Role Interpretation

Figure 3: The DBMM model allows us to uncover patterns of behavior in an email network. (a) evolving memberships for a group of nodes and (b) the characteristics associated with the roles.

transition models and define a similarity function between these vectors.

First we estimate a single transition model $\boldsymbol{T}$ for each node using the stacked model. We then compute an $n \times n$ similarity matrix using Frobenious loss between the transition matrices from the nodes. Next, we apply the classical k-means clustering algorithm to cluster the nodes by their transition matrices. Afterwards, we compute the closest rank-k approximation (k = 2 or 3) of the similarity matrix. The nodes are plotted using the low-rank approximation and labeled using the previous clustering algorithm. To reveal the structural transition pattern, we then compute the average dynamic mixed-membership for each cluster using only the nodes from that cluster.

This clustering method reveals common structural trends and patterns between nodes. For instance, this technique may group nodes together that share similar transitional patterns such as nodes with stable roles vs. nodes with more dynamic roles or nodes with high activity vs. nodes with low activity. An example is provided in Figure 4. For clarity in the visualization, we randomly selected a small subset of nodes from the 183,389 candidates and identified common transition patterns among them. The first visualization in Figure 4(a) identifies four distinct well-separated clusters of nodes with similar transition models. Figure 4(b) shows the average dynamic behavioral mixed-membership for each cluster. This visualization shows that each cluster represents a unique structural transition pattern between the nodes. The structural patterns can be interpreted using the previous role interpretation from Figure 2(c). This technique can be used for general exploratory analysis such as characterizing the patterns and trends of nodes or eventually used as a means to detect anomalies or nodes that do not fit any transition pattern.



(a) Transition Clustering

(b) Dynamic Mixed-Membership Patterns

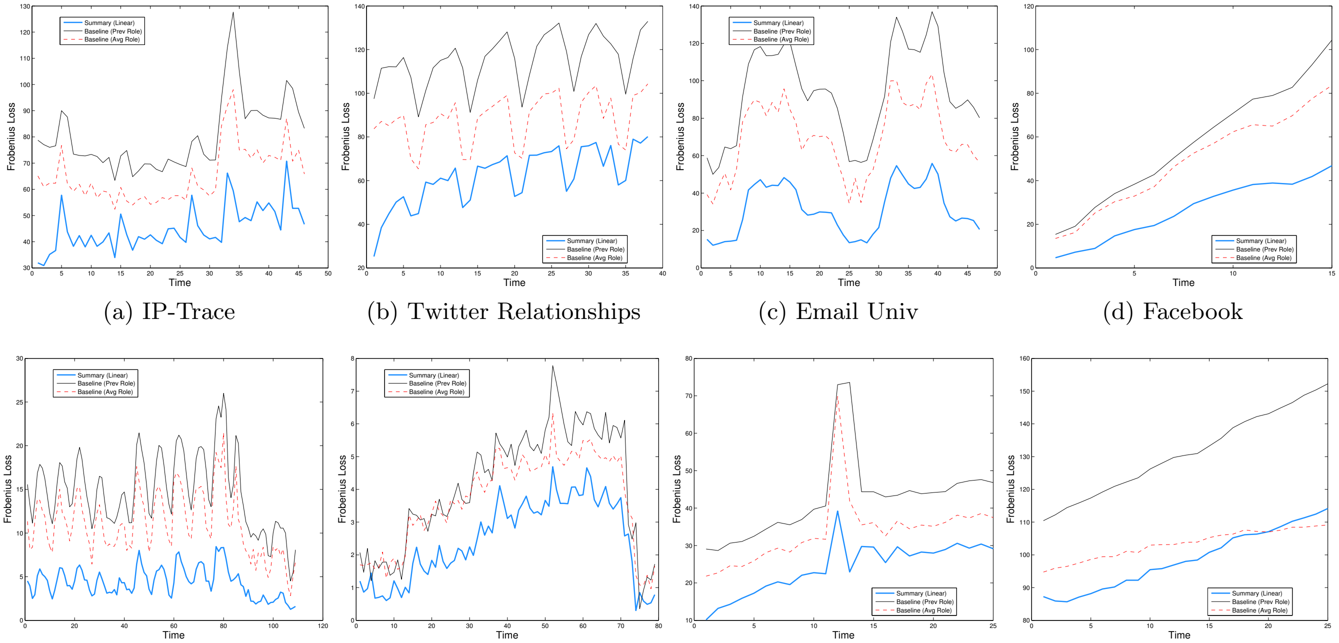
Figure 4: The DBMM model provides an intuitive means of clustering nodes that exhibit similar patterns of behavior over time. (a) identifies four distinct clusters of nodes with similar transition patterns. (b) provides a sense of the behavior of each cluster in terms of the average role-membership over time. Again, we see that DBMM captures differences in both overall static behavior (i.e., the specific roles that dominate) and in patterns of how behaviors (i.e., roles) change over time.

4. PREDICTING FUTURE BEHAVIOR

In this section, we further validate the utility of the DBMM model by demonstrating its ability to predict the future behavior of nodes. This could be useful for optimizing caches on the Web, or for improving dynamic social recommendation systems, among many others.

Models. The goal is to accurately predict $\boldsymbol{G}_{t+1}$ given $\boldsymbol{G}_{s(t)}$, the summary behavioral snapshot described in Section 2.3.2. Our primary means of predicting $\boldsymbol{G}_{t+1}$ is using our DBMM summary transition model $\boldsymbol{T}$ as follows: $\hat{\boldsymbol{G}}_{t+1} = \boldsymbol{G}_t \; \boldsymbol{T}$. We compare this summary model to two sensible baselines: *PrevRole* and *AvgRole*. PrevRole simply assigns each node the role distribution from the previous time t. That is, $\hat{\boldsymbol{G}}_{t+1} = \boldsymbol{G}_t$. AvgRole assigns each node the average role distribution over all nodes at time t. The AvgRole model can be expressed as $\hat{\boldsymbol{G}}_{t+1} = \boldsymbol{G}_t \; \boldsymbol{T}_A$ where $\boldsymbol{T}_A$ is estimated from $\boldsymbol{G}_t = [1] \; \boldsymbol{T}$. Essentially, PrevRole assumes node behavior does not change from each point in time to the next and AvgRole assumes that all nodes exhibit the average behavior of the network.

Evaluation. We consider two strategies for evaluating our predictive models: (a) compare the predicted $\hat{\boldsymbol{G}}_{t+1}$ to the true $\boldsymbol{G}_{t+1}$ using a loss function (we use the Frobenius norm) and (b) Use $\hat{\boldsymbol{G}}_{t+1}$ to predict the modal role of each node at time $t+1$ and evaluate these predictions using a multi-class



(a) IP-Trace (b) Twitter Relationships (c) Email Univ (d) Facebook

(e) Twitter Copenhagen (f) Enron (g) IMDB (h) Internet AS

Figure 5: The DBMM transition model accurately predicts future behavior of individual nodes (i.e., mixed role membership) compared to sensible baseline models.

AUC (Area Under the ROC curve) measure. We describe each of these strategies more formally below.

Frobenious Loss on $\boldsymbol{G}$. The goal here is to estimate $\boldsymbol{G}_{t+1}$ as accurately as possible. The approximation error between the estimated node memberships $\hat{\boldsymbol{G}}_{t+1} = \boldsymbol{G}_t\boldsymbol{T}_{t+1}$ and the true node memberships $\boldsymbol{G}_{t+1}$ is defined as $||\boldsymbol{G}_{t+1} - \hat{\boldsymbol{G}}_{t+1}||_F$

Structural Prediction with Multi-class AUC. This is a multi-class classification task where the true class label for node i is the modal role from the ith row of $\boldsymbol{G}_{t+1}$ (i.e., the role with maximum membership for this node). The predicted class label for node i is the modal role from the ith row of $\hat{\boldsymbol{G}}_{t+1}$.

We evaluate the predictions using a generalization of AUC extended for multi-class problems. In particular, we compute the AUC of all combinations of labels and take the mean (also known as Total AUC) [13]. The difficulty of the prediction task varies based on the number of roles discovered, complexity of the network evolution, and the type of time-evolving network (e.g., transactional vs. stable).

Results. Figure 5 demonstrate that the DBMM summary transition model is an effective predictor across the range of experiments. With few exceptions, DBMM outperforms both baselines for all data sets and timesteps. This is even true for the more complex time-evolving networks such as Twitter, email, and the IP-traces, which are more transactional with rapidly evolving network structure. For brevity, some findings were omitted, for others see [23].

In addition to validating the DBMM model, both figures offer some interesting insights into the characteristics of time-evolving networks. For example, an increase in loss over time may indicate a "concept drift" where behavior in the network has evolved to the point where the current roles can no longer adequately explain node behavior. This effect is seen most prominently in Figures 5(b), 5(d), 5(g) and 5(h). Interestingly, the drift we see in Figure 5(h) agrees with the current understanding that the underlying evolutionary process of the Internet AS is not constant, as was previously believed [19, 29]. Most notably, there is recent evidence of the Internet topology transitioning from hierarchical to a flat topological structure [7, 6].

Furthermore, the figures provide insights into behavioral anomalies, such as the spike we see in Figure 5(g). The spike in loss indicates the significant deviation of the node roles at a specific time.

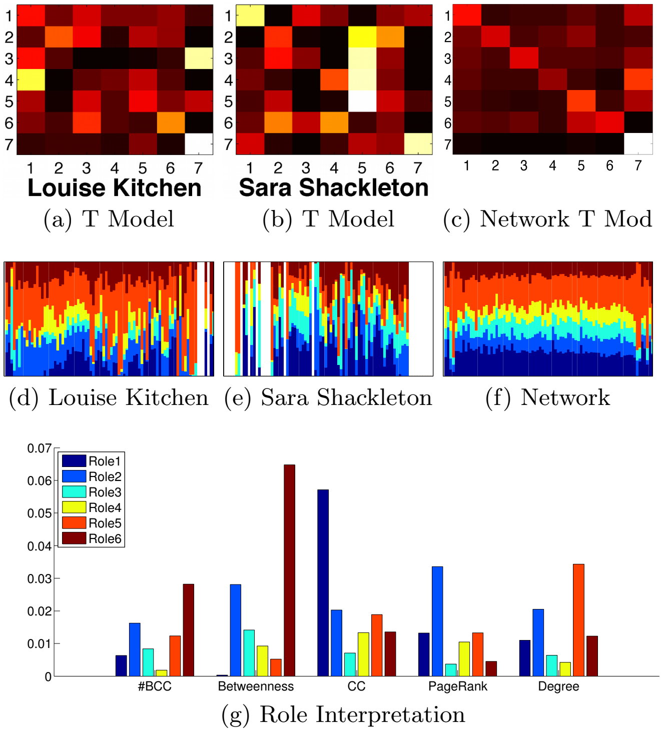
Finally, in the large Twitter Relationships network, we see seasonality among the role transitions. In particular, we find that the users generally behave significantly different over the weekends, seen by the increase in loss on these days. Intuitively, we would expect users to be tweeting about different topics and using the system in different manner than they do during the work days. The Twitter Copenhagen network captures the more locally-temporal seasonality; that is users behave differently during the daytime and the nighttime hours.

5. ANOMALOUS DYNAMIC PATTERNS

We further demonstrate the use of DBMMs for detecting anomalies in time-evolving networks. In particular, we formulate this problem with respect to identifying nodes that have *unusual structural transition patterns*. For instance, a node might transition from being a hub (i.e., a node with many people linking to it) to a node with low degree.

5.1 Node Transition Anomalies

While there are many ways to define an anomaly detection technique with respect to the DBMM model, we propose an intuitive algorithm shown in Alg. 1 that uses a node's transition model for predicting the network memberships at $t+1$. The anomaly score is the difference between the predicted network mixed-memberships and the ground-truth mixed-memberships. Therefore, the score represents the divergence of that nodes transitions from the entire network. One sim-



(a) T Model (b) T Model (c) Network T Mod

(d) Louise Kitchen (e) Sara Shackleton (f) Network

(g) Role Interpretation

Figure 6: The DBMM transition model provides an effective means of automatically discovering and visualizing nodes with anomalous temporal behavior. (a)-(b) are the transition models for two of the most anomalous nodes in the Enron email network compared to (c) the normal network transition model. (d)-(f) show the corresponding role memberships over time. (g) shows the characteristics of roles.

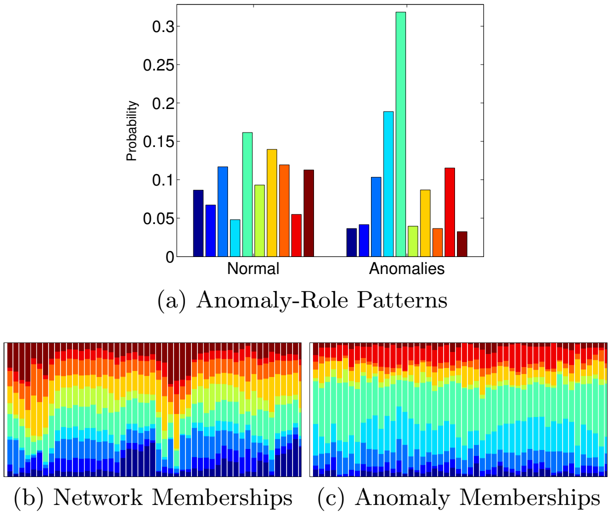
ple example is shown in Figure 6(a) where we find Louise Kitchen as having unusual behavioral transitions.

5.2 Time-varying Node Anomalies

For detecting the specific time interval in which a node has unusual behavior we use the previous method with a few subtle distinctions. The global and node models are estimated at each *timestep* (in a sort of streaming fashion) using the stacked representation with a shorter window (for leveraging past training examples). The final result is a ranked list of potential node anomalies for each timestep, shown in Figure 8. The justification for such an approach is that nodes may become anomalous or have unusual behavior only for a specified time interval. In the case of IP-communications, it is unlikely for the behavior of an IP-address to remain unusual as IP-addresses are released/expires and users are assigned entirely new IP-addresses. These types of dynamic anomalies are shown in Figure 8.

5.3 Anomalous Structural Transitions

We first interpret the roles and their temporal variation quantitatively as shown in Figure 6(g) and then provide some simple examples of nodes that have unusual behavior transitions. Intuitively, the first role represents nodes with high clustering coefficient, the second role represents mainly nodes with high pagerank, while the third and fourth roles represent some type of combination of these properties indicating a more complex structural motif that is not



(a) Anomaly-Role Patterns

(b) Network Memberships (c) Anomaly Memberships

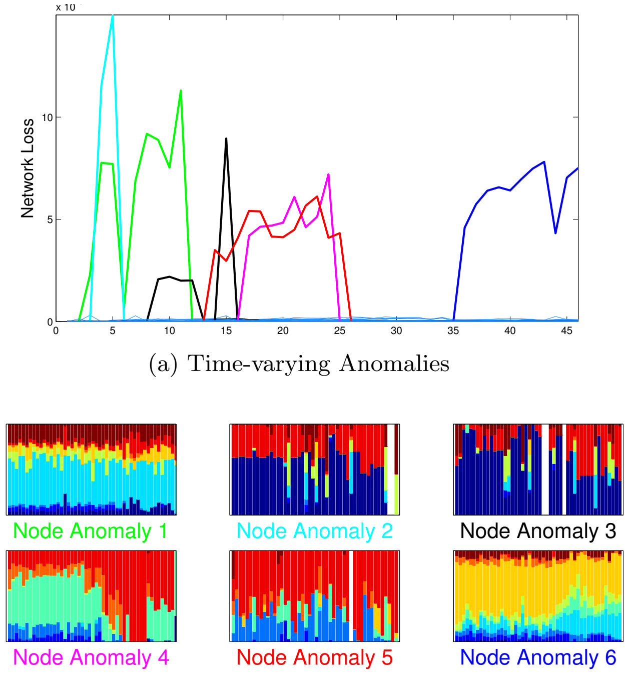
Figure 7: The DBMM anomaly detector effectively captures differences in both static and dynamic behavior in an email network. (a) shows that normal and anomalous nodes (top-100) differ in their role distribution (i.e., overall static behavior). (b)-(c) show that normal and anomalous nodes also differ in how their behavior changes over time, with anomalies exhibiting more stable behavior over time than normal.

sufficiently represented by the selected node metrics. However, the fifth role represents nodes with high degree and the sixth role represents nodes that are articulation points or that have high betweenness. Additionally, by analyzing the neighbors roles dynamically, we find that nodes with high clustering coefficient primarily are neighbors to nodes with high betweenness or high degree (this plot has been removed for brevity).

In Figure 6(e), we find Louise Kitchen, one of the Enron executives who was involved in the Fraud, as having unusual behavioral transitions. Further examination of the network transition model and the average evolution of the behavioral mixed-memberships provide further insights into his abnormal activities. In particular, there are two main role transitions ($r_3 \to r_7$ and $r_4 \to r_1$) in Louise Kitchen's transition model that are in contradiction with the network transition model shown in Figure 6(c). Furthermore, analyzing the individual changes to the mixed-memberships over time compared with the average behavioral mixed-memberships provides additional insight. For instance, the first two mixed-memberships vectors of Louise Kitchen are mainly red and then begin to deviate significantly with seemingly no un-

Algorithm 1: Anomalous Structural Transitions

```
Input: $\mathbf{G} = \{\boldsymbol{G}_t : t = 1, ..., t_{max}\}$ (evolving
       mixed-memberships)
Output: $\mathbf{x}$ (vector of anomalous scores)
for $i \leftarrow 1$ to $n$ do
    $\boldsymbol{T}^{(i)} \in \mathbb{R}^{r \times r} \leftarrow NMF(\boldsymbol{G}^{(i)}_{1:t-1}, \boldsymbol{G}^{(i)}_{2:t})$
    $\hat{\boldsymbol{G}}_{t+1} = \boldsymbol{T}^{(i)} \cdot \boldsymbol{G}_t$
    $\mathbf{x}^{(i)} = \left\| \hat{\boldsymbol{G}}_{t+1} - \boldsymbol{G}_{t+1} \right\|_F$
end
```



(a) Time-varying Anomalies

(b) Evolving memberships of the time-varying anomalies

Figure 8: The DBMM model allows us to find nodes that are anomalous for only short periods of time and normal otherwise. Such temporally local anomalies are often impossible to find using static graph analysis because brief abnormal periods are drowned out by mostly normal behavior. (a) shows examples of short lived anomalies in a computer network. (b) shows the corresponding behavior over time for each node in detail.

derlying correlation or pattern between the role transitions. Moreover, there is not any significant correlation between Louise's mixed-memberships and the average memberships at each point in time.

In addition, we also identify interesting patterns of the nodes with unusual behavioral transitions in Figure 7. In particular, the DBMM anomaly detector effectively captures differences in both static and dynamic behavior in an email network. Interestingly, in 7(b), normal users exhibit a clear cyclical pattern which indicates that normal nodes have a set of roles during the day and another set at night (which agrees with intuition). In contrast, the anomalies 7(c) have stable roles over the time that barely fluctuate.

Figure 8 also indicates that the DBMM model allows us to find nodes that are anomalous for only short periods of time and normal otherwise. Such temporally local anomalies are often impossible to find using static graph analysis because brief abnormal periods are drowned out by mostly normal behavior. 8(a) shows examples of short lived anomalies in a computer network. 8(b) shows the corresponding behavior over time for each node in detail.

Synthetic Data. In a separate set of experiments, we further validate our "unusual structural transition" anomaly detector by injecting anomalies into synthetic data (see § 3.2). Initially, the dynamics of nodes are predefined to have normal transitions between patterns (e.g., star-center to clique). Then we inject some nodes with anomalous transition behavior by randomly transitioning to an abnormal pattern which we define as star-edge to clique. For 200 repeated simulations, we achieve high accuracy (88.5%) in detecting the anomalous behavior.

Table 3: Performance Analysis of the Dynamic Behavioral Mixed-Membership Model. The dMMSB takes a day to handle 1,000 nodes [30], while our model takes only 8.44 minutes for 8,000 nodes.

Dataset	Nodes	Edges	Performance
Enron	151	50,572	117.51 sec
Twitter (Copen)	8,581	27,889	506.61 sec
Facebook	46,952	183,831	1,468.65 sec
Internet AS	37,632	505,772	1,922.85 sec
Network-Trace	183,389	1,631,824	16,138.71 sec

6. SCALABILITY AND COMPLEXITY

Most importantly, our dynamic role model is linear in the number of edges. Let n be the number of nodes, f be the number of features, r be the number of roles and t be the number of timesteps. The feature discovery is $O(t(mf + nf^2))$ [14]. For the NMF step, we use the multiplicative update method which has worst case $O(tnfr)$. The transition models is $O(tnr^2)$ using the multiplicative update method. Thus, the running time of DBMM is linear in the number of edges, specifically, $O(t(mf + nfr + nr^2))$. The time-scale t is usually small compared to the edges (even when the time-scale corresponds to minutes or seconds in the IP-trace data). A more accurate bound can be stated in terms of the maximum number of edges at any given timestep.

Our model is capable of handling realistic networks such as social and technological networks consisting of millions of nodes and edges. This is in contrast to a similar dynamic mixed-membership models that have been recently proposed such as the dMMSB [30, 9]. These models are quadratic in the number of nodes and therefore unable to scale to the realistic networks with the number of edges in the millions. Furthermore, these models have been typically used for visualizing trivial sized networks of 18 nodes up to 1,000 nodes. This is in contrast to our work where we apply DBMM not only for visualizations, but for a variety of analysis tasks using large dynamic networks.

Moreover, the dMMSB can handle 1,000 nodes in a day [30] (See page 30), while our model handles ≈8,000 nodes in 506.61 seconds (or 8 minutes and 26 seconds) shown in Table 3. We provide performance results for other larger datasets of up to 183,389 nodes and 1,631,824 edges. In all cases, even for these large networks with over a million edges, our model takes less than a day to compute and the performance results show the linearity of our model in the number of edges. For the scalability experiments, we recorded the performance results using a commodity machine Intel Core i7 @2.7Ghz with 8Gb of memory.

In addition, the proposed DBMM model is also trivially parallelizable (e.g., using Hadoop/MapReduce on Amazon EC2/Cloud) as features, roles, and transition models can be learned at each timestep independent of one another.

7. RELATED WORK

There has been an abundance of work in analyzing dynamic networks. However, the majority of this work focuses on dynamic patterns [10, 17, 16, 22, 27], temporal link prediction [8], anomaly detection [1], dynamic communities [18, 28, 11], dynamic ranking [20, 24], and many others [31, 12].

In contrast, we propose a scalable temporal behavioral model that captures the node behaviors over time and consequently learns a predictive model for how these behaviors evolve over time. Perhaps the most related work is that of [9] where they develop the dMMSB model to identify roles in the graph and how these memberships change over time. However, this type of mixed-membership model assumes a specific parametric form, which is not scalable (1,000 nodes takes a day to model), and where the groups are defined through linkage to specific nodes (in particular types of groups) rather than more general node behavior or structural properties [30]. This is in contrast to our proposed model, which is based on our intuitive behavioral representation and can be interpreted quantitatively. In addition, our model is not tied to any single notion of behavior and thus is flexible in the roles discovered and generalizable. Moreover, not only do we evaluate our model on detecting unusual behavior, identifying explainable patterns and trends, and for clustering nodes with respect to their transition patterns, but we apply our model on large real-world networks to demonstrate its scalability. To the best of our knowledge, our proposed model is the first scalable dynamic mixed-membership model capable of identifying explainable patterns and trends on large networks.

8. CONCLUSIONS

We proposed a dynamic behavioral mixed-membership model for large networks and used it for identifying interesting and explainable patterns and trends. Moreover, we demonstrated its scalability on a variety of real-world temporal networks and provided striking performance results. The experiments have shown the scalability, flexibility, and effectiveness of our model for identifying interesting patterns, detecting unusual structural transitions, and predicting the future structural changes of the network and individual nodes.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This research was also supported by ARO and NSF under contract numbers W911NF-08-1-0238, IIS-1017898, IIS-0916686. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of ARO, NSF or the U.S. Government.

9. REFERENCES

[1] J. Abello, T. Eliassi-Rad, and N. Devanur. Detecting novel discrepancies in communication networks. In *ICDM*, 2010.

[2] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *SIGKDD*, 2007.

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD*, 2006.

[4] M. Cha, A. Mislove, and K. P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *WWW*, 2009.

[5] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *KDD*, 2006.

[6] A. Dhamdhere and C. Dovrolis. Ten years in the evolution of the internet ecosystem. In *SIGCOMM*, 2008.

[7] A. Dhamdhere and C. Dovrolis. The internet is flat: modeling the transition from a transit hierarchy to a peering mesh. In *In CoNEXT*, page 21, 2010.

[8] D. Dunlavy, T. Kolda, and E. Acar. Temporal link prediction using matrix and tensor factorizations. *TKDD*, 5(2):10, 2011.

[9] W. Fu, L. Song, and E. Xing. Dynamic mixed membership blockmodel for evolving networks. In *ICML*, pages 329–336. ACM, 2009.

[10] M. Gotz, J. Leskovec, M. McGlohon, and C. Faloutsos. Modeling blog dynamics. In *ICWSM*, 2009.

[11] D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *ASONAM*, 2010.

[12] H. Habiba, Y. Yu, T. Berger-Wolf, and J. Saia. Finding spread blockers in dynamic networks. In *ASONAM*, 2008.

[13] D. Hand and R. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

[14] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos. It's Who You Know: Graph Mining Using Recursive Structural Features. In *SIGKDD*, pages 1–10, 2011.

[15] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos. RolX: Role Extraction and Mining in Large Networks. In *KDD*, 2012.

[16] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. *TWEB*, 1(1):1–39, 2007.

[17] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*, 2005.

[18] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. Tseng. Analyzing communities and their evolutions in dynamic social networks. *TKDD*, 3(2):8, 2009.

[19] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, et al. The Internet AS-level topology: three data sources and one definitive metric. *ACM SIGCOMM CCR*, 36(1):17–26, 2006.

[20] J. O'Madadhain, J. Hutchins, and P. Smyth. Prediction and ranking algorithms for event-based network data. *SIGKDD Explorations*, 7(2):30, 2005.

[21] R. Pan and J. Saramaki. Path lengths, correlations, and centrality in temporal networks. *arXiv:1101.5913*, 2011.

[22] S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In *VLDB*, 2005.

[23] R. Rossi, B. Gallagher, J. Neville, and K. Henderson. Modeling temporal behavior in large networks: A dynamic mixed-membership model. In *LLNL-TR-514271*, 2011.

[24] R. Rossi and D. Gleich. Dynamic pagerank using evolving teleportation. In *WAW*, pages 126–137. LNCS 7323, 2012.

[25] R. Rossi, L. McDowell, D. Aha, and J. Neville. Transforming graph data for statistical relational learning. *Journal of Artificial Intelligence (JAIR)*, 2012.

[26] R. Rossi and J. Neville. Time-evolving relational classification and ensemble methods. In *PAKDD*, 2012.

[27] J. Sun, C. Faloutsos, S. Papadimitriou, and P. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD*, 2007.

[28] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *SIGKDD*, pages 677–685. ACM, 2008.

[29] X. Wang, X. Liu, and D. Loguinov. Modeling the Evolution of Degree Correlation in Scale-Free Topology Generators. In *INFOCOM*, 2007.

[30] E. Xing, W. Fu, and L. Song. A state-space mixed membership blockmodel for dynamic network tomography. *Ann. Appl. Stat.*, 4(2):535–566, 2010.

[31] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, 2011.