Multi-mode Network Data Synthesis: A Network Algebra and Systems Approach

Valenzuela, Michael L.[a,*], Rozenblit, Jerzy W.[a], Breiger, Ronald L.[b]

[a] *University of Arizona, Electrical and Computer Engineering, 1230 E. Speedway Blvd, Tucson, AZ*
[b] *University of Arizona, School of Sociology, 1145 E. South Campus Dr., Tucson, Arizona*

Abstract

A fundamental problem for network analysis is the discovery of cohesive subgroups, cliques, blocks, or communities within networks. One of the current obstacles in multi-mode networks research is the gap between the ground-truth and observed data. In this paper, we discuss an innovative approach to data synthesis, using the theory of network semigroup algebras with linear system theory, to produce generative, stochastic, nonlinear models. Since our motivating and illustrative example is a human-centric cyber-security problem, we model human-human, human-machine, and machine-machine interactions. We also apply and extend a multi-mode community detection and show both it and simulation correctly.

Keywords: Simulation, Network Algebra, Multi-Mode Networks, Linear System Theory, Cyber-Security

*Corresponding author
Email addresses: mvalenz@email.arizona.edu (Valenzuela, Michael L.), jr@ece.arizona.edu (Rozenblit, Jerzy W.), breiger@email.arizona.edu (Breiger, Ronald L.)

This work has been supported by NSF Grant No: CNS-1347075: Human-Centric Predictive Analytics of Cyber-Threats: a Temporal Dynamics Approach.

1. Introduction

A fundamental problem for network analysis is the discovery of cohesive subgroups, cliques, blocks, or communities within networks. An explosion of developments followed the work of Girvan and Newman [1], bringing community detection to the attention of the mathematics and statistical physics communities [2]. If we extend networks from social and biological realms to encompass mixed networks, including but not limited to computer, evidence, time, and social networks, then we call this construct a multi-mode network (*cf.* De Domenico et al. [3]). Combining these two facets, the problem of multi-mode community detection emerges. Multi-mode networks have wide applicability, encompassing all the aforementioned networks.

Recently, cyber criminals have already taken a multi-mode approach to penetrate secure systems [4, 5, 6]. As such, the study of multi-mode networks is one promising venue for human-centric cyber-security. Most cyber-security approaches use graph theory [7], packet tagging/marking [8], signatures [9], deep packet inspection [10], and anomaly detection [11, 12]. However, Oh et al. [13] have focused on the social aspects of security. This motivates our investigation into multi-mode networks for security purposes.

Nevertheless, one of the current drawbacks to multi-mode research and analysis is the lack of data for which the ground-truth is known. By ground-truth, we mean what is truly happening as opposed to the data, which may be incomplete, noisy, filled with red herrings, include purposeful mistruths, or any combination of these. Such incomplete knowledge hinders the development of new techniques that could unearth the ground-truth. So how does one deal with an unknown ground-truth?

Consider the following four remedies to this lack of ground-truth information. First, one could attempt to collect the real world multi-mode data. However, this would be cost prohibitive, potentially unethical, and would fail to provide the ground-truth. Second, one could orchestrate a scenario and collect all information. This approach would probably produce unrealistic data or be cost

prohibitive. Third, one could generate synthetic data using existing theory. Fourth, the previous three techniques could be combined in a hybrid approach.

In this paper, we discuss an innovative approach to data synthesis using existing theory. Since our motivating example is a human-centric cyber-security problem, we combine the theory of network semigroup algebras with linear system theory to produce a generative, stochastic, nonlinear model of human-human, human-machine, and machine-machine interactions. Moreover, our data synthesis allows the multi-mode network to evolve over time, allowing us to investigate analysis methods that exploit temporal information. We investigate the technique discussed in Melamed et al. [14] with the synthetic data generated here. This technique ignores temporal information, but we extend it by taking a temporal difference and examine these results as well.

Section 2 reviews existing data synthesis techniques, multi-mode representations and analysis, network semigroup algebras, and a brief review of linear system theory. We combine linear system theory with network semigroup algebra to arrive at a generative, stochastic, nonlinear model in Section 3 before discussing an illustrative example in Section 4. We conclude with a discussion of future work in Section 6.

2. Background

We briefly review five subjects in this section. After covering the justification, recent history, and ethics of data synthesis, we discuss network representations and their extensions into multi-mode networks. Then, we rehash prior work using network semigroup algebras. Since we combine multi-mode networks and semigroup algebras with discrete state-spaces, we recall the basics of discrete state-spaces. The last bit of background we examine is a recent multi-mode partitioning methodology.

2.1. Data Synthesis

Data synthesis can be seen as the outcome of a simulation. Simulations and modeling have long been used to replace or supplement real data and make pre-

dictions. Consider how physics provides mathematical models of nature; these models helped send humanity to the moon. Researchers sometimes optimize a real system by optimizing a model of the system. Traffic simulations have helped scientists test/optimize traffic control [15]. Discrete Event Simulations (DES) of Discrete Event System Specifications (DEVS) date back to the work of Zeigler [16]. However, so far simulations within social networks have mostly been limited to the propagation of ideas [17], emotions [18], and diseases [19]. On the other hand, Kumar et al. [20] studied the evolution of social networks with an emphasis on the growth and migration of star networks. While there exists some research into the evolution of multi-mode networks [21], it is focused on clustering and community detection, rather than multi-mode network simulation.

One more concern with regards to the generation and use of synthetic data: ethics. Never attempt to present data generated from a model as data from the real world. The National Science Foundation (NSF) states that data fabrication and falsification are forms of research misconduct.[1] One should always make every effort to ensure synthetic data is understood by all as the result of an imperfect model. Always overtly disclose the use of simulations and data synthesis. The methods discussed in this research should never be used to generate synthetic data that could even be misinterpreted as genuine multi-mode data. George E. P. Box captured this idea when he said "essentially, all models are wrong, but some are useful." Our method provides useful data, but it should not be mistaken as the real thing.

2.2. Network Representations

There exist many common representations for networks in general: adjacency lists, adjacency matrices, and incidence matrices. The term adjacency means the structure describes the relationship between nodes. The term, incident, is used to describe relationships between nodes and vertices. Lists tend to be more

[1] See `http://www.nsf.gov/oig/_pdf/presentations/session.pdf`

efficient for representing sparsely connected networks and matrices for dense networks. Adjacency matrices are commonly used in the literature [22, 23, 1, 24, 2, 14, 3]. However, when dealing with multiple modes, adjacency matrices need some adaptation. De Domenico et al. [3] use multi-layer adjacency tensors. Kivelä et al. [25] provide a review of several more multilayer network extensions.

We adopt a matrix of matrices approach. This is similar to the representation in De Domenico et al. [3], yet avoids tensorial notation (for better or for worse). This representation is interchangeable with block-matrices, allowing us to use block notation similar to that presented in Melamed et al. [14]:

$$\begin{pmatrix} \underline{\underline{Z_{1,1}}}(n) & \underline{\underline{Z_{1,2}}}(n) & \underline{\underline{Z_{1,3}}}(n) \\ \underline{\underline{Z_{2,1}}}(n) & \underline{\underline{Z_{2,2}}}(n) & \underline{\underline{Z_{2,3}}}(n) \\ \underline{\underline{Z_{3,1}}}(n) & \underline{\underline{Z_{3,2}}}(n) & \underline{\underline{Z_{3,3}}}(n) \end{pmatrix}$$

where any double underlined variable is a matrix *per sé*. We use this double underline notation throughout the rest of this paper. Similarly, singly underlined variables are column vectors. Any variable without any underlines is a scalar. For asymmetric matrices, the entity of the row has a connection/influence on an entity of the corresponding column when there is a connection. This is in agreement with using right multiplication to chain on the next relationship or action.

We have one final note about this matrix of matrices perspective. Each block-matrix can be interpreted as a different kind of network. Carley [26] gives an example using four modes. The matrix involving interactions between people is a social network. Interactions between people and knowledge, Carley calls a knowledge network. In our approach, interactions between computers occur in a computer network; interactions between computers and people are called a user network.

2.3. Network Algebras

Network algebras provide a means to describe compound relations. Pioneering work was done by Boorman and White [27]. Breiger and Pattison [22]

say that if there are two types of connections, say $\underline{\underline{L}}$ and $\underline{\underline{M}}$ (for loans and marriages between families), then we ought to consider $\underline{\underline{L}}^2$, $\underline{\underline{L}}\,\underline{\underline{M}}$, $\underline{\underline{M}}\,\underline{\underline{L}}$, $\underline{\underline{M}}^2$, etc. In this scenario $\underline{\underline{M}}\,\underline{\underline{L}}$ connects family A to family C, such that a member of A is married to a spouse in some family (B) containing someone who makes a loan to C. The semigroup is just the clamped[2], multiplicative closure of all types of connection matrices. The semigroup describes all possible interactions between these connections. Now if we consider a new type of connection/relationship, which may even be a temporal update to an existing type of connection, it makes sense to consider at least the simple compound relations described by the existing semigroup; one family may be more likely to give a loan to another family that shares a marriage connection. Martino and Spoto [24] share this same idea of using compound relations to form new relations.

Breiger and Pattison [23] discuss algebraic models for social networks in general. This includes algebras from networks with types of relationships, including the aforementioned semigroups. They present role algebras as a local network analogue to semigroups, which are in turn algebras with a single binary operation, and partial algebras as algebras with restrictions on repeated operations. In multi-mode networks, the definition of "local" is ambiguous, leading to both mode roles (semi-local roles) and individual roles (fully local roles). Moreover, sometimes we require partial algebras for computational feasibility, especially if one goes beyond binary multiplication.

2.4. Discrete State-Spaces

Discrete linear systems may be described in state-space form as [28, chap. 3.4][29, chap 8.1]:

$$\underline{x}(n+1) = \underline{x}(n)\,\underline{\underline{A}}(n) + \underline{\underline{B}}(n)\,\underline{u}(n), \tag{1}$$

$$\underline{y}(n+1) = \underline{x}(n)\,\underline{\underline{C}}(n) + \underline{\underline{D}}(n)\,\underline{u}(n) + \underline{N}(n), \tag{2}$$

[2]The result of clamped matrix multiplication must return a matrix where all connections are in the interval [0,1], sometimes resulting in nonlinear behavior.

where

- $\underline{x}(n)$ is the state vector at time n,
- $\underline{y}(n)$ is the output vector,
- $\underline{\underline{A}}(n)$ is a matrix describing the linear system,
- $\underline{\underline{B}}(n)$ is an input mixing matrix,
- $\underline{\underline{C}}(n)$ is a matrix mapping from the state vector to its contribution of the output,
- $\underline{\underline{D}}(n)$ is an output mixing matrix,
- $\underline{\mathcal{N}}(n)$ is a noise vector,
- $\underline{u}(n)$ input to the system, and
- n is the discrete time.

Typically, $\underline{\underline{B}}(n)$, $\underline{\underline{D}}(n)$, and $\underline{u}(n)$ are ignored unless one is attempting to control the system. One can readily see that these terms may be ignored when $\underline{u}(n)$ is a vector of zeros. Standard linear system theory rarely explicitly states noise ($\underline{\mathcal{N}}$). One could also include a noise term influencing the state. However, we chose to ignore noise in the feedback loop to better match our extension to multi-mode networks.[3]

2.5. A multi-mode partitioning methodology

Melamed et al. [14] use a multi-mode eigenspectrum approach to partition multi-mode networks. It is the combination of the multi-mode networks set forth by Fararo and Doreian [30] and the eigenspectrum approach of Newman [31]. Essentially, intra-mode connections are ignored, each cross-mode matrix is replaced with its modularity matrix (the connection matrix subtract the null

[3]Even multi-mode networks may have noise in their feedback loop, but for simplicity we ignore it in this paper.

model), and entities are partitioned using the eigenvectors from the resulting matrix (of matrices).

Let $\underline{\underline{\underline{\underline{A'}}}}$ be a binary adjacency matrix and $\underline{\underline{\underline{\underline{P}}}}(\underline{\underline{\underline{\underline{A'}}}})$ be the null model matrix describing the probability of connections. Then each modularity matrix is

$$\underline{\underline{\underline{\underline{B'}}}}\left(\underline{\underline{\underline{\underline{A'}}}}\right) = \underline{\underline{\underline{\underline{A'}}}} - \underline{\underline{\underline{\underline{P}}}}\left(\underline{\underline{\underline{\underline{A'}}}}\right). \tag{3}$$

$\underline{\underline{\underline{\underline{P}}}}\left(\underline{\underline{\underline{\underline{A'}}}}\right)$ is a matrix representing the null model. It is usually the outer product of row sums with column sums normalized by the total number of connections,

$$\begin{aligned} P\left(\underline{\underline{\underline{\underline{A'}}}}\right) &= \frac{\underline{A'}_{i,\Sigma}\,\underline{A'}^{\intercal}_{\Sigma,j}}{\sum_l \sum_k A'_{l,k}} \\ \underline{A'}_{\Sigma,j} &= \sum_k A'_{k,j} \\ \underline{A'}_{i,\Sigma} &= \sum_k A'_{i,k}{}^{\intercal}, \end{aligned} \tag{4}$$

or sometimes a matrix filled with the average connection strength,

$$P\left(\underline{\underline{\underline{\underline{A'}}}}\right)_{i,j} = \frac{\sum_l \sum_k A'_{l,k}}{\sum_l \sum_k 1}.$$

Compute $\underline{\underline{\underline{\underline{P}}}}(\underline{\underline{\underline{\underline{\cdot}}}})$ for all cross-mode adjacency matrices. The next step to partition the multi-mode network is to zero out the block-diagonal. For concreteness, assume we are working with three symmetric modes:

$$\begin{pmatrix} \underline{\underline{\underline{\underline{0}}}} & \underline{\underline{\underline{\underline{B}}}}\left(\underline{\underline{\underline{\underline{Z_{1,2}}}}}(n)\right) & \underline{\underline{\underline{\underline{B}}}}\left(\underline{\underline{\underline{\underline{Z_{1,3}}}}}(n)\right) \\ \underline{\underline{\underline{\underline{B}}}}\left(\underline{\underline{\underline{\underline{Z_{1,2}}}}}(n)\right)^{\intercal} & \underline{\underline{\underline{\underline{0}}}} & \underline{\underline{\underline{\underline{B}}}}\left(\underline{\underline{\underline{\underline{Z_{2,3}}}}}(n)\right) \\ \underline{\underline{\underline{\underline{B}}}}\left(\underline{\underline{\underline{\underline{Z_{1,3}}}}}(n)\right)^{\intercal} & \underline{\underline{\underline{\underline{B}}}}\left(\underline{\underline{\underline{\underline{Z_{2,3}}}}}(n)\right)^{\intercal} & \underline{\underline{\underline{\underline{0}}}} \end{pmatrix}$$

Taking this result, its eigenvectors may be used to partition the network. Using the eigenvector having the largest eigenvalue, the community can be partitioned into two groups using the signs of the eigenvector. Naturally, this can be recursed to partition into more than two communities. Newman [31, 32] recommends repeating this process as long as it improves the modularity and provides a

thorough discussion about how to get subsequent partitions. Taking the entities in one or both communities, further partition them using the signs of the second largest eigenvector.

Melamed et al. [14] originally tested their technique on a small didactic example. However, to fully verify their method, they had to simulate thousands of networks. Their simulated networks were based primarily on three factors: size, density, and probabilities of ties between communities. One may question the realism of these simulated networks. Thus, this serves as further motivation for us to create better simulated multi-mode networks. Moreover, we will investigate this method, including extensions to temporal multi-mode networks.

3. Data Synthesis Approach

The primary thrust of our work is to create better realistic multi-mode networks through an extension of discrete state-spaces to multi-mode network semigroup algebras. Our proposed method allows us to embed ground-truth, watch it evolve, and watch an apparent-truth evolve as well. We first extend linear system theory to single mode network systems in Section 3.1. Then, in Section 3.2, we extend this to multi-mode networks.

3.1. Single Mode Network Data Synthesis

We start this section by introducing one possible network system extension to linear system theory, explaining and justifying the three primary modifications. The bulk of this section discusses the semantics of each variable and their relation to network algebra. Again, as mentioned previously, for asymmetric matrices the entity of the row has a connection/influence on an entity of the corresponding column when there is a connection. If the reader prefers the opposite, take the transpose of all matrix expressions.

$$\underline{\underline{X}}(n+1) = \underline{\underline{X}}(n)\,\underline{\underline{A}}(n) + \underline{\underline{B}}(n)\,\underline{\underline{U}}(n), \tag{5}$$

$$\underline{\underline{Y}}(n+1) = \underline{\underline{X}}(n) \odot \underline{\underline{C}}(n) + \underline{\underline{D}}(n) \odot \underline{\underline{U}}(n) + \underline{\underline{N}}(n), \tag{6}$$

$$\begin{aligned} &\text{s.t.} \\ 0 \leq X(n)_{i,j} &\leq 1 \\ 0 \leq Y(n)_{i,j} &\leq 1 \\ \underline{\underline{A}}(n) &\in S \end{aligned} \tag{7}$$

where $\odot$ is the Hadamard (elementwise) product, and

- $\underline{\underline{X}}(n)$ is the ground-truth/state *matrix* at time n for one type of relation,
- $\underline{\underline{Y}}(n)$ is the *apparent*-truth/state,
- $X(n)_{i,j}$ is the element at the i^{th} row, j^{th} column of $\underline{\underline{X}}(n)$,
- $Y(n)_{i,j}$ is the element at the i^{th} row, j^{th} column of $\underline{\underline{Y}}(n)$,
- $\underline{\underline{A}}(n)$ is a (stochastic) matrix describing new interactions,
- $\underline{\underline{B}}(n)$ is an input mixing matrix,
- $\underline{\underline{C}}(n)$ is a ground-truth masking matrix,
- $\underline{\underline{D}}(n)$ is an output mixing matrix,
- $\underline{\underline{N}}(n)$ is a noise matrix,
- $\underline{\underline{U}}(n)$ is the input to the system, and
- n is the discrete time.
- S is the semigroup of (other) ground-truth relationships.

We will ignore $\underline{\underline{B}}(n)$, $\underline{\underline{D}}(n)$, and $\underline{\underline{U}}(n)$ in this paper, saving their use for future work. One can see that the influence of $\underline{\underline{B}}(n)$ and $\underline{\underline{D}}(n)$ disappear when $\underline{\underline{U}}(n)$ is a matrix of zeros.

There exist three key differences between typical linear system theory and our multi-mode network extension. In the multi-mode network systems, the ground-truth state, the apparent state, and input are all matrices. This is reasonable since the multiplication between $\underline{\underline{X}}(n)$ and $\underline{\underline{A}}(n)$ models the events the network algebra predicts and the resulting state natively reflects the relationship matrix. Second, (6) replaces standard matrix multiplication with elementwise multiplication. We justify this with the following two facts: (A) the Hadamard product makes it easy to model hidden information and (B) it seems strange that observed connections would be a linear combination of the ground-truth connections. The third difference is that (5) and (6) are made nonlinear through the constraints that each connection is clamped to the range $[0, 1]$.

$\underline{\underline{X}}(n)$ represents the ground-truth relationship matrix for a *single* kind of relation. By ground-truth, we mean that this represents the true state of the world. This is as opposed to the apparent-truth, $\underline{\underline{Y}}(n)$. $\underline{\underline{Y}}(n)$ is the relationship matrix as observed, including noise and missing information. Thus, the difference between $\underline{\underline{X}}(n)$ and $\underline{\underline{Y}}(n)$ represents the information gap between what is *actually* happening and *is thought* to be happening. Ignoring input into the system,

$$\underline{\underline{X}}(n) = \underline{\underline{X}}(0)\,\underline{\underline{A}}(0)\,\underline{\underline{A}}(1)\,\ldots\,\underline{\underline{A}}(n-1),$$

where $\underline{\underline{A}}(n)$ for all n also comes from ground-truth relationships.

Compositions of typed relations (interactions) may lead to new relations [24]. A temporal update is one such new relation. Thus, $\underline{\underline{A}}(n)$ is used to update the current state. Specifically, $\underline{\underline{A}}(n)$ is a matrix from the semigroup formed from ground-truth relationships. These relationships may be latent or clearly visible; our generative model is indifferent toward whether these additional types of relationships are hidden. $\underline{\underline{A}}(n)$ is rarely selected from the full semigroup algebra, but rather is selected from a partial algebra consisting of only a few

repeated operations. This is to model only so many events happening within a single temporal update. For example, if up to three events were allowed to happen in a single temporal update, then $\underline{\underline{A}}(n)$ would be randomly selected from the partial algebra generated using only three matrix multiplications of ground-truth relationships (including the identity matrix). This random selection makes our model stochastic.

How should one select $\underline{\underline{A}}(n)$? It depends on the desired level of realism and the amount of data one has. With minimal data and realism, all matrices in the partial semigroup algebra (PSA) have equal probability. For a moderate level of fidelity, probabilities of fundamental types of interactions can be multiplied together to estimate the probabilities of all matrices in the PSA. Extending the marriage and loan example in [22], if the probabilities interactions involving a loan relationship, a marriage relationship, and no relationship are $P(\underline{\underline{L}}) = 0.2$, $P(\underline{\underline{M}}) = 0.1$, and $P(\underline{\underline{I}}) = .7$, then the probability of $\underline{\underline{M}}\,\underline{\underline{L}}$ would be $0.2 \cdot 0.1 = 0.02$. For the most realistic scenarios when data is abundant, $\underline{\underline{A}}(n)$ may be selected using a Markov process. This would allow the probability of $\underline{\underline{M}}\,\underline{\underline{L}}$ to differ from $\underline{\underline{L}}\,\underline{\underline{M}}$.

What if the $\underline{\underline{X}}(n)$ is not a network at all, but rather some physical process. Those with a background in quantum physics or quantum computation may note the similarity of (5) to the methods of using group theory to evolve quantum states. Quantum information may be modeled to evolve discretely in time through the use of group theory [33, chap. 2.3]. Using this analogy, $\underline{\underline{\mathrm{A}}}(\mathrm{n})$ may be a Hadamard quantum gate. This is only an imperfect metaphor, since "kets" are also vectors and the equations typically do not model interference in the same fashion. However, this still provides some physical justification for our simulation model even when the state represents something other than a network.

$\underline{\underline{U}}(n)$ allows us to model the connections we can influence. $\underline{\underline{B}}(n)$ primarily serves as a convoluting matrix, limiting the controllability of this stochastic dynamic nonlinear system. Moreover, control of such a system would in general be complicated. We conjecture that only rarely will the system ever be control-

lable. The nonlinear and non-curvilinear nature (clamping connections to the range $[0, 1]$) will make linear control theory inapplicable and optimal control difficult. Moreover, some entities are sentient, making this a game theoretic problem. We believe that numerical optimization techniques may make certain desired outcomes more likely, but due to the stochastic nature these will be computationally expensive. To further complicate matters, the role of $\underline{\underline{D}}(n)$ in (6) is uncertain. Perhaps it models our ability to deceive ourselves in our power to control the system? As such, we leave the entire subject of control to future work.

3.2. Multi-mode Network Data Synthesis

Now we use a matrix of matrices representation for multi-mode networks. The block-diagonal matrices represent interactions within a single mode, while the block off-diagonal matrices represent interactions between modes. Using the matrix of matrices representation, the state (either ground-truth or observed) would look like:

$$\begin{pmatrix} \underline{\underline{Z_{1,1}}}(n) & \underline{\underline{Z_{1,2}}}(n) & \underline{\underline{Z_{1,3}}}(n) \\ \underline{\underline{Z_{2,1}}}(n) & \underline{\underline{Z_{2,3}}}(n) & \underline{\underline{Z_{2,3}}}(n) \\ \underline{\underline{Z_{3,1}}}(n) & \underline{\underline{Z_{3,2}}}(n) & \underline{\underline{Z_{3,3}}}(n) \end{pmatrix}.$$

Here, $\underline{\underline{Z_{i,j}}}(n)$ corresponds to an $\underline{\underline{X}}(n)$ or $\underline{\underline{Y}}(n)$. The block-diagonal matrices are updated using (5) for the ground-truth and (6) for the observed state, just as previously discussed. Updating the block off-diagonal matrices presents a new challenge since they are generally rectangular.

We see two possible approaches for updating these off-diagonal matrices. The first approach uses only (5) and (6) for each sub-matrix, but $\underline{\underline{A_{i,j}}}(n)$ and $\underline{\underline{C_{i,j}}}(n)$ must be appropriately sized matrices. Meanwhile, the second approach still uses (5) and (6), but only for the block-diagonal elements. The block off-diagonal elements are updated using matrices that map between intra-mode interactions and inter-mode connections. For example,

$$\underline{\underline{Z_{i,j}}}(n+1) = \underline{\underline{Z_{i,j}}}(n) + \underline{\underline{A_{i,i}}}(n)\underline{\underline{M_{i,j}}}(n),$$

where $\underline{\underline{\underline{\underline{A_{i,i}}}}}(n)$ is an $n \times n$ state matrix and $\underline{\underline{\underline{\underline{M_{i,j}}}}}(n)$ is an $n \times m$ inter-mode mapping matrix. Here, inner-product multiplication is natural. It allows entities in one mode to correspond to entities in another mode.

Some researchers may want to use the first approach. We offer the following comments and describe behaviors from this approach. In this approach, one may choose to update any number of cross-mode matrices in a single unit of time. For concreteness, let us examine three modes. The cross-mode matrices would be

$$\begin{pmatrix} \underline{\underline{0}} & \underline{\underline{\underline{\underline{Z_{1,2}}}}}(n) & \underline{\underline{\underline{\underline{Z_{1,3}}}}}(n) \\ \underline{\underline{\underline{\underline{Z_{2,1}}}}}(n) & \underline{\underline{0}} & \underline{\underline{\underline{\underline{Z_{2,3}}}}}(n) \\ \underline{\underline{\underline{\underline{Z_{3,1}}}}}(n) & \underline{\underline{\underline{\underline{Z_{3,2}}}}}(n) & \underline{\underline{0}} \end{pmatrix}.$$

We will first consider symmetric connections. If symmetry is desired, allow $\underline{\underline{\underline{\underline{Z_{i,j}}}}}(n) = \underline{\underline{\underline{\underline{Z_{j,i}}}}}(n)^{\intercal}$. This would also ensure all block-diagonal matrices are square, but still potentially of different sizes. Each cross-mode matrix would correspond to an instance of an $\underline{\underline{X}}(n)$ matrix. But, $\underline{\underline{X}}(n)$ depends on $\underline{\underline{A}}(n-1)$, and $\underline{\underline{A}}(n-1)$ is vague in this cross-mode scenario. $\underline{\underline{A}}(n)$ must be a square matrix for repeated operations to make sense, but cross-mode matrices will be rectangular in general. Using the convention that subsequent interactions are right multiplied, $\underline{\underline{A}}(n)$ must be a square matrix having as many columns as the cross-mode matrix. Thus, $\underline{\underline{A}}(n)$ represents the entities from the mode of the corresponding column. To preserve symmetry, after updating $\underline{\underline{\underline{\underline{Z_{i,j}}}}}$ use $\underline{\underline{\underline{\underline{Z_{i,j}}}}}(n) = \underline{\underline{\underline{\underline{Z_{j,i}}}}}(n)^{\intercal}$ to update the corresponding cross-mode matrix. Another side effect of this approach is that, without further modification, the connections for each cross-mode adjacency matrix will update independently of each other. For our purposes, this behavior is undesirable.

Extending the first approach to handle asymmetric connections offers additional flexibility. If $\underline{\underline{\underline{\underline{X_{i,j}}}}}(n)$ were rectangular, then $\underline{\underline{\underline{\underline{A_{i,j}}}}}(n)$ must have different dimensions than $\underline{\underline{\underline{\underline{A_{j,i}}}}}(n)$. In general, this will drive asymmetry, which is sometimes desirable. However, this increases the independence of cross-modes further, a behavior we are trying to avoid. This brings us to our second approach

for updating the cross-mode matrices.

The second approach, which we use in Section 4, is to propagate the updates from the matrices along the block-diagonal. Suppose we update $\underline{\underline{Z_{2,2}}}(n)$ using the method described in Section 3.1. Denote the interaction matrix for the second mode as $\underline{\underline{A_{2,2}}}(n)$ and the visible state as $\underline{\underline{Y_{2,2}}}(n)$. Propagating these changes symmetrically, we have:

$$\begin{pmatrix} \cdot & \underline{\underline{Z_{1,2}}}(n+1) & \cdot \\ \underline{\underline{Z_{1,2}}}(n+1)^\intercal & \underline{\underline{Z_{2,2}}}(n+1) & \underline{\underline{Z_{2,3}}}(n+1) \\ \cdot & \underline{\underline{Z_{2,3}}}(n+1)^\intercal & \cdot \end{pmatrix}, \tag{8}$$

where

$$\begin{aligned} \underline{\underline{Z_{2,2}}}(n+1) &= \underline{\underline{Y_{2,2}}}(n+1) \\ \underline{\underline{Z_{1,2}}}(n+1) &= \underline{\underline{Z_{1,2}}}(0) + \underline{\underline{M_{1,2}}}(n)\, \underline{\underline{A_{2,2}}}(0)\, \underline{\underline{A_{2,2}}}(1) \ldots \underline{\underline{A_{2,2}}}(n) \\ \underline{\underline{Z_{2,3}}}(n+1)^\intercal &= \underline{\underline{Z_{2,3}}}(0)^\intercal + \underline{\underline{M_{2,3}}}(n)\underline{\underline{A_{2,2}}}(0)\, \underline{\underline{A_{2,2}}}(1) \ldots \underline{\underline{A_{2,2}}}(n) \end{aligned} \tag{9}$$

$$\tag{10}$$

and $\cdot$ is a placeholder for matrices which stay the same. $\underline{\underline{M_{i,j}}}(n+1)$ is an event mapping of the same dimensions as $\underline{\underline{Z_{i,j}}}(n+1)$. $\underline{\underline{M_{i,j}}}(n+1)$ is a rectangular matrix describing how interactions within the column mode may affect the row mode. The cross-modes above and below the updated mode use their event mapping matrices directly. The cross-modes to the left and right come from the transpose of the vertically updated elements. This means they use the transpose of the mapping, for left matrix multiplication, and transposes of the actions:

$$\begin{aligned} \underline{\underline{Z_{2,3}}}(n+1) &= \left(\underline{\underline{Z_{2,3}}}(n+1)^\intercal\right)^\intercal \\ &= \underline{\underline{Z_{2,3}}}(0) + \left(\underline{\underline{M_{2,3}}}(n)\underline{\underline{A_{2,2}}}(0)\, \underline{\underline{A_{2,2}}}(1) \ldots \underline{\underline{A_{2,2}}}(n)\right)^\intercal \\ &= \underline{\underline{Z_{2,3}}}(0) + \underline{\underline{A_{2,2}}}(n)^\intercal\, \underline{\underline{A(n-1)_{2,2}}}^\intercal \ldots \underline{\underline{A_{2,2}}}(0)^\intercal \underline{\underline{M_{2,3}}}(n)^\intercal. \end{aligned}$$

This has the desirable property that a single interaction within one mode produces dependencies between across modes.

We conclude this section with a few more notes about $\underline{\underline{M_{i,j}}}(n)$. $\underline{\underline{M_{i,j}}}(n)$ may optionally be stochastic, which is the primary reason we gave it a temporal index. Asymmetry can be introduced by allowing $\underline{\underline{M_{i,j}}}(n) \neq \underline{\underline{M_{j,i}}}(n)^\intercal$. The sub-matrices may also be asymmetric, even if the multi-mode matrix of matrices is symmetric in terms of its block matrices.

4. Illustrative Example

Here, we provide a simple illustrative example of how our method can be used to simulate multi-mode networks over time. In this concrete example, we will create an initial state and evolve it into a new state. The initial conditions/connections are explained in a story like fashion. The multi-mode network dynamics are also specified allowing the network to evolve through the processes in Section 3.1 and 3.2. After that, we analyze the results of the simulation using standard methods and the partitioning method discussed in Melamed et al. [14].

4.1. Initialization

Consider the following pilot scenario. Let there be three people: Alice, Bob, and Carol. The ground-truth is that Bob, through Alice, hears about Carol. Bob decides to compromise Carol's computer in order to steal her identity. In the process, Bob was seen using a library computer. However, since Bob could not compromise Carol's computer directly, he installs a worm/virus/script on one of the library computers to infect computers it connects to, spreading the script. Eventually, his virus infects Carol's computer when she uses a library computer to log into her home computer.

The initial intra-mode matrices are listed below. Starting with the intra-

mode matrix for people, each row/column corresponds to Alice, Bob, Carol:

$$\underline{\underline{PEP}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

In the initial intra-mode matrix for evidence, each row/column corresponds to the library, the virus, Bob's fingerprints, and a candy bar wrapper (as a red-herring):

$$\underline{\underline{EVI}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Effectively, no evidence is initially connected. The initial computer intra-matrix is shown below, with each row/column corresponding to the first library computer, the second library computer, and Carol's computer.

$$\underline{\underline{COMP}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

that is, all computers are initially disconnected from each other.

Now, consider the cross-mode matrices. For clarity, these are expressed as $\underline{\underline{\text{mode 1} \times \text{mode 2}}}$. The $\times$ symbol stands for "cross," as in cross-mode. The first initial cross-mode matrix we discuss is the people–evidence matrix. Its rows are people in the same order as in the $\underline{\underline{PEP}}$ matrix and the columns are evidences in the same order as the $\underline{\underline{EVI}}$ matrix:

$$\underline{\underline{PEP \times EVI}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

This says that Bob has ties to the library, the virus, and his fingerprints. It also shows that Carol uses the library, but she is unrelated to the fingerprints.

The next initial cross-mode matrix describes the relationships between the evidence and computers. The rows correspond to the evidence in the same order as it is mentioned in $\underline{\underline{EVI}}$ and columns correspond to the two library computers and Carol's computer:

$$\underline{\underline{EVI \times COMP}} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

This tells us the first two computers are in the library. This also says that Bob's fingerprints are on the first library computer. The second row says *initially* no computers have any viruses on them. The virus is a dynamic element, not yet present at the initialization of the simulation.

The cross-mode matrix for the people cross computers is:

$$\underline{\underline{PEP \times COMP}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

This means the third computer is Carol's computer. Thus, the initial setup is:

$$\begin{pmatrix} \underline{\underline{PEP}} & \underline{\underline{PEP \times EVI}} & \underline{\underline{PEP \times COMP}} \\ \underline{\underline{PEP \times EVI}}^{\intercal} & \underline{\underline{EVI}} & \underline{\underline{EVI \times COMP}} \\ \underline{\underline{PEP \times COMP}}^{\intercal} & \underline{\underline{EVI \times COMP}}^{\intercal} & \underline{\underline{COMP}} \end{pmatrix}, \tag{11}$$

which expands to

$$\left(\begin{array}{ccc|cccc|ccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right). \tag{12}$$

Now, for simplicity, we only consider dynamics of the computers. We ignore who logs into Carol's home computer; we only care if/when it is logged into and what the current state is when that occurs. Consider two possible interactions: computer 1 connects to computer 2 and computer 2 connects to Carol's computer. The first interaction, corresponding to a potential $\underline{\underline{A_{3,3}}}$ event is:

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{13}$$

The second interaction is captured by:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}. \tag{14}$$

Lastly, we let the identity matrix be part of the semigroup algebra, corresponding to nothing happening. Technically, this means our semigroup algebra is also a monoid. Our monoid contains five possible outcomes: nothing happens, just (13), just (14), (13) $\cdot$ (14), and (14) $\cdot$ (13). Notice that only (13) $\cdot$ (14),

interpreted as computer 1 connects to computer 2 followed by computer 2 connects to Carol's computer, produces a connection from the first computer to Carol's computer. If the order of connections is reversed, then Carol's computer remains safe. This is because we are modeling short-term/asymmetric connections. These matrices would be symmetric if we wanted to model persistent connections.

The above is only sufficient for intra-mode simulation. We still need mapping matrices to use (8) and (9), to propagate these changes across modes. They may be deterministic or stochastically selected. In this example, allow there to be just one mapping matrix:

$$\underline{\underline{M_{2,3}}}(n) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{15}$$

The rows correspond to the evidence matrix and the columns correspond to the computers. This means the first computer has is infected by the computer virus. If the computers go unused, Bob's virus just sits there. However, interaction (13) will cause the virus to spread to computer 2. If interaction (14) occurs first, the virus will remain quiescent.

Likewise, we will need another mapping matrix to describe how interactions within the computer mode will influence the human mode. Bob, being the author of the virus, will become associated with the infected computers. This makes the $\underline{\underline{M_{1,3}}}(n)$ matrix:

$$\underline{\underline{M_{1,3}}}(n) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{16}$$

The ground-truth is that Bob becomes associated with the first computer after he infects it; the rest of the connections will follow.

4.2. Simulation Results

Using (8) and (9) with $\underline{\underline{A_{3,3}}} = (13) \times (14)$, $\underline{\underline{M_{2,3}}}(n)$ described by (15), and $\underline{\underline{M_{1,3}}}(n)$ described by (16), a single temporal update makes the following changes. $\underline{\underline{COMP}}$, $\underline{\underline{EVI \times COMP}}$, and $\underline{\underline{PEP \times COMP}}$ become

$$\underline{\underline{COMP}}(1) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\underline{\underline{EVI \times COMP}}(1) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and

$$\underline{\underline{PEP \times COMP}}(1) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

This makes the ground-truth matrix after the first update (corresponding to at least two interactions):

$$\left(\begin{array}{ccc|cccc|ccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right). \tag{17}$$

The difference between this matrix and the initial state is:

$$\left(\begin{array}{ccc|cccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right). \tag{18}$$

4.3. Analysis

We investigate a few analysis methods here on the initial connection matrix, the evolved connection matrix, and the matrix of changes. This includes measures of centrality, betweenness, and the modularity technique mentioned in Melamed et al. [14]. We only analyze the ground-truth matrix to demonstrate the network structure exists before noise is added and some connections are masked.

Melamed's, Breiger's, and West's technique produced the following results shown in Table 1. Initially, with two partitions this technique works well to isolate Bob, the virus, his fingerprints, and the computer he installed the virus on. However, the method fails to extract Bob with the virus in the evolved system. The maximum modularity with three partitions splits Bob, the virus, his fingerprints, and the computer he installed the virus on into two groups. Again, the pattern is lost in the evolved connection matrix.

Does this mean there is something wrong with the analysis technique or something wrong with the simulation? Melamed et al. [14] use Newman's definition of expected values [31]; the probability that a connection exists is equal to the outer product of probabilities that a connection is present for the row and a connection is present for the column. That is, Newman used the

Table 1: Partitioning by method in Melamed et al. [14]

Matrix	Modularity	Entity Partitions
Initial	0.2083	**(1)** Alice, Carol, library, candy bar, computer 2, computer 3 **(2)** Bob, virus, fingerprints, computer 1
Final	0.1429	**(1)** Carol, library, virus, computer 2, computer 3 **(2)** Alice, Bob, fingerprints, candy bar, computer 1
Initial	0.1771	**(1)** Alice, Carol, library, candy bar, computer 2, computer 3 **(2)** Bob, virus **(3)** fingerprints, computer 1
Final	0.1488	**(1)** Alice, Bob, fingerprints, candy bar, computer 1 **(2)** virus, computer 3 **(3)** Carol, library, computer 2

null model described by (4). The method depends on sparsity. The evolved multi-mode network shows Bob has connections to almost everything, ruining his significance, and subsequently places him in the same group as the candy bar. Replacing the null model with the uniform model, (3), produces the data in Table 2. This alternative partitioning works better on this illustrative example, highlighting Bob, the virus, and the evidence before and after simulation.

Moreover, when we tried the technique described in Melamed et al. [14], we encountered a problem analyzing the δ matrix. The evolved state was not sufficiently different from the original state. This problem was compounded using Newman's null model, (4). The independence model produced only zero eigenvalues (and hence all partitions considered equal). However, using the uniform null model, (3), produced positive results. These are summarized in Table 2. The Δ matrix describes activity. Melamed's, Breiger's, and West's method, with a uniform null model, when applied to changes in the network, appears to effectively isolate activity.

The uniform null model makes a lot of sense when analyzing changes in network connectivity. It follows from the maximum energy principle. It is robust, working even when the changes in connections form an ultra sparse matrix. Lastly, empirically, at least on this pilot problem, it appears to work. The question remains whether the uniform null is appropriate to use in the absence of temporal data. Investigation of other alternative null models is saved for future work.

Betweenness metrics are shown in Table 3. Again, the Δ Matrix (changes that happened during simulation) highlight Bob, the virus, and the computers. Notice that compared to the final matrix, the Δ Matrix shows a larger betweenness value for the virus and a lower value for Bob. In some regards, this reflects that stopping Bob and stopping the virus would have similar effects on the network.

Tables 4, 5 and 6 display the eigenvector, Katz, and degree centrality measures. Katz centrality measure uses a decay parameter less than or equal to the inverse of the largest (by magnitude) eigenvalue. Since the largest eigenvalue of

Table 2: Melamed et al. [14] partitioning with a uniform null model

Matrix	Modularity	Entities
Initial	0.3924	**(1)** Bob, library, virus, fingerprints, computer 1, computer 2 **(2)** Alice, Carol, candy bar, computer 3
Final	0.3214	**(1)** Bob, library, virus, fingerprints, computer 1, computer 2, computer 3 **(2)** Alice, Carol, candy bar
Δ	0.7083	**(1)** Bob, virus, computer 1, computer 2, computer 3 **(2)** Alice, Carol, library, fingerprints, candy bar
Initial	0.5069	**(1)** Bob, library, virus, fingerprints, computer 1, computer 2 **(2)** candy bar **(3)** Alice, Carol, computer 3
Final	0.3690	**(1)** Bob, library, virus, fingerprints, computer 1, computer 2, computer 3 **(2)** candy bar **(3)** Alice, Carol
Δ	0.7083	**(1)** Bob, virus, computer 1, computer 2, computer 3 **(2)** candy bar **(3)** Alice, Carol, library, fingerprints

Table 3: Betweenness metric

Entity	Betweenness		
	Initial Matrix	Final Matrix	Δ Matrix
Alice	5.333	1.117	0
Bob	24	19.10	1.500
Carol	16.67	2.750	0
Library	28	4.900	0
Virus	0	1.333	1.500
Fingerprints	2.667	0	0
Candy bar	0	0	0
Computer 1	3.333	4.567	0.6667
Computer 2	0	0.9167	0.6667
Computer 3	0	4.317	0.6667

all three matrices is 5.284, limiting the attenuation factor to 0.1892, we exhibit the Katz centrality with a decay parameter of $\alpha = 0.15$. In all three metrics, Bob and the computers become more important in the evolved state. Contrasting these three metrics, only the eigenvector centrality always weights the first computer (originally infected) more central than Carol's computer. The other two metrics swap the emphasis between these computers after the state evolves. This swap is also present in the Δ matrix.

Naturally, many more tests could be conducted. De Domenico et al. [34] discuss centrality in interconnected multilayer networks. This metric will be investigated in future work.

5. Computational Complexity Analysis

The simulation methodology discussed in Section 3.2 has a computational complexity bounded by $\mathcal{O}(t \cdot s_m^2 \cdot s_t)$, where s_m is the maximum size of all modes, s_t is the total size of the multi-mode network, and t is the number of time steps. The slowest operation is updating the inter-mode matrices. In the worst case

Table 4: Eigenvector centrality

Entity	Eigenvector Centrality		
	Initial Matrix	Final Matrix	Δ Matrix
Alice	0.3207	0.1572	0
Bob	0.4920	0.4997	0.4502
Carol	0.3615	0.1738	0
Library	0.5055	0.3459	0
Virus	0.1849	0.3617	0.4502
Fingerprints	0.2983	0.2263	0
Candy bar	0	0	0
Computer 1	0.3020	0.4700	0.5661
Computer 2	0.1899	0.3382	0.4204
Computer 3	0.1358	0.2416	0.3122

Table 5: Katz centrality, $\alpha = 0.15$

Entity	Katz Centrality		
	Initial Matrix	Final Matrix	Δ Matrix
Alice	1.050	1.951	0
Bob	1.657	5.573	0.8025
Carol	1.293	2.480	0
Library	1.695	3.612	0
Virus	0.6454	3.967	0.8025
Fingerprints	0.9992	2.145	0
Candy bar	0.1765	0.1765	0
Computer 1	1.005	3.582	0.5408
Computer 2	0.6521	3.835	0.7719
Computer 3	0.5811	4.489	1.038

Table 6: Degree Centrality

Entity	Degree Centrality		
	Initial Matrix	Final Matrix	Δ Matrix
Alice	3	3	0
Bob	5	8	3
Carol	4	4	0
Library	5	5	0
Virus	2	5	3
Fingerprints	3	3	0
Candy bar	1	1	0
Computer 1	3	5	2
Computer 2	2	5	3
Computer 3	2	6	4

scenario, each event mapping matrix (sized s_k by s_m) is multiplied by an s_m by s_m matrix. This means updating a single inter-mode matrix is bounded by $s_m^2 \cdot s_k$. However, this must happen for all modes. Noting that $\sum_k s_k = s_t$, a single time step takes $\mathcal{O}(s_m^2 \cdot s_t)$. Thus, the run time is bounded by $\mathcal{O}(t \cdot s_m^2 \cdot s_t)$.

6. Conclusions and Future Work

Multi-mode networks extend social, biological, computer, and evidence network. Moreover, multi-mode networks easily capture temporal data with time slices. However, automated processing of multi-mode networks is still in its infancy. Before we can even begin to test analysis tools on multi-mode networks, we need both data and the ground-truth. While multi-mode data is becoming more available, it remains scarce. Scarcer still is knowledge of what actually happened. In an effort to overcome these obstacles, we formulated a multi-mode simulation framework built upon semigroup network algebras, linear system theory, and group theory.

We developed a multi-mode network simulation framework where the ground-truth can evolve according to a semigroup algebra, but the same methodology also happens to allow discrete linear systems to evolve, too. In this general framework, the output is the ground-truth plus noise and some connections/evidence are hidden. Due to the similarity between linear system theory and our multi-mode simulation, it would be interesting to see if control methods of linear systems may be generalized to at least partially work on multi-mode networks. This in turn would open up the future study on control of multi-mode networks.

We tested the technique described in Melamed et al. [14] and found that a different null model works better on our synthetic data. The uniform null model allowed us to better partition the active agents in an illustrative example. Moreover, the uniform null model makes a lot of sense when looking at how connections change over time due to its empirical efficacy, robustness to ultra sparse changes, and the fact that it follows from the maximum entropy principle. It was this kind of test we were hoping synthetic data can aid in future multi-mode analysis research.

Kivelä and Porter [35] discuss isomorphisms in multilayer networks, across a variety of network types including temporal networks. This leads to another interesting extension to our own work. What if we could simulate backward in time as well as forward in time? The reasons to rewind a network are myriad, ranging from answering how a network evolved into its current state to rewinding from a goal/target network to see how easily reachable it is. If the updates applied the multi-mode network are invertible, then in a probabilistic sense, the resulting networks are all isomorphisms of each other.

[1] M. Girvan, M. E. J. Newman, Community Structure in Social and Biological Networks, Proceedings of the national academy of sciences 99 (12) (2002) 7821–7826, doi:10.1073/pnas.122653799, web. 7 July 2015.

[2] M. A. Porter, P. J. Mucha, M. E. J. Newman, A. J. Friend, Community structure in the United States House of Representatives, Physica A:

Statistical Mechanics and its Applications 386 (1) (2007) 414–438, ISSN 0378-4371, doi:http://dx.doi.org/10.1016/j.physa.2007.07.039, URL http://www.sciencedirect.com/science/article/pii/S0378437107007844.

[3] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M. A. Porter, S. Gómez, A. Arenas, Mathematical Formulation of Multilayer Networks, Phys. Rev. X 3 (2013) 041022, doi:10.1103/PhysRevX.3.041022, URL http://link.aps.org/doi/10.1103/PhysRevX.3.041022.

[4] T. N. Jagatic, N. A. Johnson, M. Jakobsson, F. Menczer, Social Phishing, Commun. ACM 50 (10) (2007) 94–100, ISSN 0001-0782, doi:10.1145/1290958.1290968, URL http://doi.acm.org/10.1145/1290958.1290968.

[5] M. Workman, Gaining Access with Social Engineering: An Empirical Study of the Threat, Information Systems Security 16 (6) (2007) 315–331, doi:10.1080/10658980701788165, URL http://dx.doi.org/10.1080/10658980701788165.

[6] M. Huber, S. Kowalski, M. Nohlberg, S. Tjoa, Towards automating social engineering using social networking sites, in: International Conference on Computational Science and Engineering, 2009 (CSE'09), vol. 3, IEEE, 117–124, 2009.

[7] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, L. Chang, Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach, in: Wireless Communications and Networking Conference, 2005 IEEE, vol. 2, ISSN 1525-3511, 1193–1199 Vol. 2, doi:10.1109/WCNC.2005.1424678, 2005.

[8] K. Kim, J. Kim, J. Hwang, IP traceback with sparsely-tagged fragment marking scheme under massively multiple attack paths, Cluster Computing 16 (2) (2013) 229–239, ISSN 1386-7857, doi:10.1007/s10586-011-0186-3, URL http://dx.doi.org/10.1007/s10586-011-0186-3.

[9] Y. Tang, S. Chen, Defending against Internet worms: a signature-based approach, in: Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies., vol. 2, ISSN 0743-166X, 1384–1394, doi:10.1109/INFCOM.2005.1498363, 2005.

[10] S. Kumar, J. Turner, J. Williams, Advanced algorithms for fast and scalable deep packet inspection, in: Architecture for Networking and Communications systems, 2006. ANCS 2006. ACM/IEEE Symposium on, 81–92, 2006.

[11] A. Patcha, J.-M. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, Computer Networks 51 (12) (2007) 3448–3470, ISSN 1389-1286, doi:http://dx.doi.org/10.1016/j.comnet.2007.02.001, URL http://www.sciencedirect.com/science/article/pii/S138912860700062X.

[12] P. Düssel, C. Gehl, P. Laskov, J.-U. Bußer, C. Störmann, J. Kästner, Cyber-Critical Infrastructure Protection Using Real-Time Payload-Based Anomaly Detection, in: E. Rome, R. Bloomfield (Eds.), Critical Information Infrastructures Security, vol. 6027 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, ISBN 978-3-642-14378-6, 85–97, doi:10.1007/978-3-642-14379-3_8, URL http://dx.doi.org/10.1007/978-3-642-14379-3_8, 2010.

[13] O. Oh, M. Agrawal, H. Rao, Information control and terrorism: Tracking the Mumbai terrorist attack through twitter, Information Systems Frontiers 13 (1) (2011) 33–43, ISSN 1387-3326, doi:10.1007/s10796-010-9275-8, URL http://dx.doi.org/10.1007/s10796-010-9275-8.

[14] D. Melamed, R. L. Breiger, A. J. West, Community Structure in Multi-Mode Networks: Applying an Eigenspectrum Approach, Connections: Official Journal of the International Network for Social Network Analysts 33 (1) (2013) 18–23.

[15] S. Sen, K. L. Head, Controlled optimization of phases at an intersection, Transportation science 31 (1) (1997) 5–17.

[16] B. Zeigler, Theory of Modeling and Simulation, Wiley Interscience, New York, first edn., 1976.

[17] E. Abrahamson, L. Rosenkopf, Social Network Effects on the Extent of Innovation Diffusion: A Computer Simulation, Organization Science 8 (3) (1997) 289–309, doi:10.1287/orsc.8.3.289, URL http://dx.doi.org/10.1287/orsc.8.3.289.

[18] V. Goel, Facebook Tinkers With Users' Emotions in News Feed Experiment, Stirring Outcry, Online, URL http://www.nytimes.com/2014/06/30/technology/facebook-tinkers-with-users-emotions-in-news-feed-experiment-stirring-outcry.html?_r=0, 2014.

[19] S. Eubank, H. Guclu, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, N. Wang, Modelling disease outbreaks in realistic urban social networks, Nature 429 (6988) (2004) 180–184, ISSN 0028-0836, URL http://dx.doi.org/10.1038/nature02541.

[20] R. Kumar, J. Novak, A. Tomkins, Structure and Evolution of Online Social Networks, in: P. S. Yu, J. Han, C. Faloutsos (Eds.), Link Mining: Models, Algorithms, and Applications, Springer New York, ISBN 978-1-4419-6514-1, 337–357, doi:10.1007/978-1-4419-6515-8_13, URL http://dx.doi.org/10.1007/978-1-4419-6515-8_13, 2010.

[21] L. Tang, H. Liu, J. Zhang, Z. Nazeri, Community Evolution in Dynamic Multi-mode Networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, ACM, New York, NY, USA, ISBN 978-1-60558-193-4, 677–685, doi:10.1145/1401890.1401972, URL http://doi.acm.org/10.1145/1401890.1401972, 2008.

[22] R. L. Breiger, P. E. Pattison, Cumulated Social Roles: The Duality Duality of Persons and Their Algebras, Social Networks 8 (3) (1986) 215–256.

[23] R. L. Breiger, P. E. Pattison, Algebraic models for social networks, 7, Cambridge University Press, Cambridge England ; New York, NY, USA, 1993.

[24] F. Martino, A. Spoto, Social Network Analysis: A brief theoretical review and further perspectives in the study of Information Technology., PsychNology Journal 4 (1) (2006) 53–86.

[25] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, M. A. Porter, Multilayer networks, Journal of Complex Networks 2 (3) (2014) 203–271.

[26] K. M. Carley, Dynamic Network Analysis, in: R. Breiger, K. M. Carley, P. Pattison (Eds.), Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, Committee on Human Factors, National Research Council, National Academies Press, Washington, DC, 133–145, 2003.

[27] S. A. Boorman, H. C. White, Social Structure from Multiple Networks: II. Role Structures, American Journal of Sociology 86 (1976) 1384–1446, URL http://www.jstor.org/stable/2777009.

[28] F. Szidarovszky, A. T. Bahill, Linear Systems Theory, chap. Characterization of Systems: Discrete Systems, CRC press, Boca Raton, FL, 2 edn., 144–145, 1997.

[29] G. F. Franklin, J. D. Powell, M. L. Workman, Digital Control of Dynamic Systems, chap. Design Using State-Space Methods, Ellis-Kagle Press, 3 edn., 280, 1998.

[30] T. J. Fararo, P. Doreian, Tripartite structural analysis: Generalizing the Breiger-Wilson formalism, Social Networks 6 (2) (1984) 141–175, ISSN 0378-8733, doi:http://dx.doi.org/10.1016/0378-8733(84)90015-7, URL http://www.sciencedirect.com/science/article/pii/0378873384900157.

[31] M. E. J. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Sciences 103 (23) (2006) 8577–8582.

[32] M. E. J. Newman, Networks: An Introduction, chap. 11.9 Division into more than two groups, Oxford Univ Press, 2010.

[33] I. Djordjevic, Quantum information processing and quantum error correction: an engineering approach, Academic press, 2012.

[34] M. De Domenico, A. Solé-Ribalta, E. Omodei, S. Gómez, A. Arenas, Ranking in interconnected multilayer networks reveals versatile nodes, Nature communications 6 (6868).

[35] M. Kivelä, M. A. Porter, Isomorphisms in Multilayer Networks, pre-print doi:http://arxiv.org/abs/1506.00508, URL `http://arxiv.org/abs/1506.00508`, arXiv 1506.00508 physics.soc-ph.