CARNEGIE MELLON UNIVERSITY

Distinguishing between different mechanisms of network evolution using network motifs and machine learning

by

Manojit Nandi

A thesis submitted in partial fulfillment for the
degree of Bachelors of Science

in the
Social and Decision Sciences Department
Advised by
Russell Golman

April 25, 2014

Declaration of Authorship

I, MANOJIT NANDI, declare that this thesis titled, 'DISTINGUISHING THE EFFECTS OF CONTAGIONS IN SOCIAL NETWORKS' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________

Date: ______________________________

CARNEGIE MELLON UNIVERSITY

Abstract

Social and Decision Sciences Department

Bachelors of Science

by Manojit Nandi

In this thesis, I will introduce two methodological tools for understanding the evolution of social networks. Using a mathematical representation of social networks based on covariance matrices, I demonstrate machine learning algorithms can use these representations of networks to properly classify networks based on their evolution mechanism with high accuracy. I also show the over-expression of particular network motifs can be used to distinguish between different network evolution processes.

Acknowledgements

This thesis was produced with guidance from Michele Tumminello and Barnabas Poczos. Michele Tumminello serves as the advisor of this thesis, and Barnabas Pocozs is my faculty advisor for the Machine Learning Senior Project. This work would not be possible without the help of all of these people. In addition, I would like to thank Russell Golman for serving as the faculty stand-in for this senior thesis.

Senior Honors Thesis

Introduction

The *dynamic evolution of networks* -how links form and dissolve in networks over time- is one of the earliest problems studied in network science. Sociologists and anthropologists study the formation and evolution of social networks because it provides an understanding of how societies form along with providing an explantion for observed phenomena in emperical social networks. In their famous experiment, Milgram and Travers showed the average distance between any two individuals in a social network is about 6 people, leading to the development of the idea that human society experiences a small-world network [1]. Granovetter showed network structures with a prevalence for weak ties reinforce the diffusion of new information throughout the network as acquaintances are less likely to be exposed to the same information as those within a person's close circle of friends [2]. More recently, Kuhn et. al showed the structure of scientific research citation networks provides a framework for understanding the transmission of scientific memes, information and key phrases that are learned through imitation [3]. Pentland argues how a deeper understanding of the relationship between network evolution and the spread of ideas could be used to develop high-information communites designed to optimize the potential for economic and scientific innovation [4].

The earlist models in this field assumed a fixed number of nodes in the network and provide probablistic interpretations of how edges form. The Erdos-Renyi model is one of the first probablistic models of network growth. In this model, they assume a fixed number of nodes in the network, and with some probability p, an edge is included between each pair of nodes, so the formation of each edge is independent of the formation of other edges [5]. While this model's simplicity allows for the derivation of certain mathematical properties, this model does not accurately portray the growth of real-word networks because the formation of edges in real-world social networks are not independent of one another.

The Watts-Strogatz model was developed to address the criticism of the Erdos-Renyi model. Given a fixed number of nodes N, a mean degree value K, and some probability value p, the Watts-Strogatz model starts with each node connected to K other nodes. For each edge, the model removes it with probability p and adds another edge in the graph, a process known as re-wiring. This model produces networks with properties found in real-world networks, such as hubs of high local clustering, but these models often have an unrealistic distribution of node degrees [6].

The Barabasi-Albert model is the first evolving model of network formation incorporating the preferential attachment property. *Preferential attachment* is the positive feedback cycle observed in real-world networks in which nodes with high degree values are more likely to form edges with new nodes in the network, a phenomena commonly referred to the rich-club effect. The Barabasi-Albert model generates networks where the degree distribution follows a power law distribution of the form $Pr[\text{Degree} = k] = k^{-\gamma}$ for some constant γ [7].

While there has been work in modeling the evolution of a particular network, there has not been much development in methodology to compare the evolution of different networks. For example, if we consider Network A and Network B, there is no statistical test we can run to determine if the processes governing the evolution of Network A differ significantly from the processes governing the evolution of Network B. In this thesis, we explore two different methodologies to distingush between different network evolution mechanisms. We develop two network evolution mechanisms based on the idea of latent homophily and social leveraging, a concept we derive from the idea of social contagions, and we demonstrate that different network motifs are over-expressed in networks produced by the two mechanisms. In addition, using a mathematical representation of social networks based on Krylov Subspaces, we show this representation can be used as features in a support vector machine to classify networks based on their growth mechanism.

Related Work

More recent work in modeling network evolution includes adding a fitness function on top of the Barabasi-Albert model, so nodes present during the creation and early stages of the network evolution are more likely to dominate the network growth process in later stages [7]. To study the evolution of real-world networks, Fleury et. al created static snapshots of the network at different times, recorded various properties, such as clustering coefficent, number of nodes, length of shortest, of each network, and performed a time series analysis to find patterns in the growth of these properties[8].

Other similar work involves modeling the evolution of communities, or sub-graphs of the network that are strongly connected, using a percolation algorithm to forecast the community's future growth based on its past growth [9].

However, these methodolgies consider only local properties of the social network and ignore the rest of the network. Methodologies using the entire graph rather than localized properties are relatively new. Felizi et al [10] use spectral deconvolution to identify direct edges in a directed network where information spread follows under closure of transitivity. Choi designs a statistical test to test for coordination in social network using the entirity of the network where the null hypothesis, lack of coordination, corresponds to latent homophily [11] .

We studied network evolution from a joint approach. The mathematical representation of networks using the Krylov Subspaces captures the spread of information in the entire network, so we can observe how information diffusion pathways change on a global scale. To complement this, the network motifs also us to identify key localized patterns that are indicative of a particular network evolution mechanism. Together, this joint approach allows us to consider how networks evolve on a global scale and how they evolve on a local scale.

Network Motifs

Network Motifs are localized sub-graphs within a social netork that re-occur at a rate that is significantly higher than what we would expect by random chance. The threshold rate for significance is determined by standard hypothesis testing for some given α value. Current network science research shows motifs reflect functional processes that occur in complex networks [12]. Recently, Conway showed network motifs can be used to model and understand the evolution mechanisms driving the growth of political networks[13]. From this work, we wish to show that social networks in which social leveraging is the primary underlying mechanism driving the growth of the network display different network motifs from those networks in which latent homophily is the primary mechanism driving network growth.

We considered only three-node motifs because all higher-order motifs can be expressed as combinations of three-node substructures. Many metrics in social network analysis consider only triples of nodes. For example, the clustering coefficient of a network is calculated as $Clustering = \frac{3 \times NumTrianges}{NumTriples}$, where *NumTriangles* is the number of closed triangles, or sets of nodes A, B, C that form a three-clique [14]. Network motifs are indexed in the *mfinder* Network Motif dictionary [15], and all three-node motifs are presented below. The number above each motif represents its index number in the Motif dictionary.

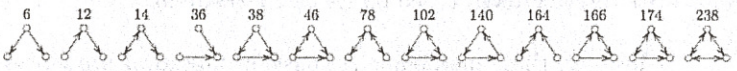

FIGURE 1: All 3-node network motifs

Shalizi and Thomas [16] showed that social contagion and latent homophily are confounded in observational social network data. We used these ideas of social contagion and latent homophily to design two network evolution mechanisms which govern network growth by similar processes.

Social Leveraging Motifs

In social network literature, *social contagions* are defined as ideas, beliefs, or behaviors spread throughout a network in a manner similar to the spread of diseases [17]. Like diseases, the propagation of social contagion occurs through contact between individuals in the network. Individuals are said to influence the behaviors of those they interact with. From this notion of social contagion, we define *social leveraging* as the process by which a node uses its current neighbors to grow more neighbors in a particular direction. A real world example of this would be professional networking in which a person asks his colleagues to introduce him to a very important person in the network. This way, the person uses his current connections to build future connections. If person A has the potential to influence person B, and person B has the potential to influence person C, then person A can leverage his connection with person B to create an opportunity to influence person C, and thereby form a link to person C. Because of this, we expect to observe *triadic closure* - the increased likelihood of individuals to form links if they share common neighbors- denoted by the presence of closed triangles in the network. In social leveraging, because potential pathways to influence are unidirectional, the closure of any triple of nodes in the network favors unidirectional links. In the example above, because there is a pathway for person A to potentially influence person C, but no pathway for person C to influence person A, then the formation of a directed edge from person A to person C is more likely than a directed edge from person C to person A.

Latent Homophily Motifs

Latent homophily is the increased likelihood that individuals with similar behaviors are more likely to form links in a network. Whereas social contagions implies observed links between individuals result in changes in behavior, latent homophily implies behavior results in changes in observed links between individuals. Latent homophily implies the formation of new edges between person A and person C can be attributed to their likelihood to attend similar events or have similar friends, so they have more opportunities to meet one another. For example, if Alice and Bob are both interested in animal rights, then they are likely to meet one another at an animal rights rally. After connecting with one another, Alice and Bob are likely to discuss and share the latest news about animal rights and related topics. If Alice is friends with Elsa, and Bob is friends with Elsa, and all three are interested in animal rights, then Elsa serves as a conduit for the exchange of ideas between Alice and Bob. Elsa and Alice could have a discussion about animal rights, and afterwards, Elsa tells Bob about some of the things she and Alice discussed. As a result, Bob is indirectly influenced by Alice's ideas. Conversely, Elsa could tell Alice about some of the things she and Bob discussed, so Alice is indirectly influenced by Bob. Therefore, Alice has the potential to influence Bob through Elsa, and Bob has the potential to influence Alice through Elsa in a symmetric manner.

In general, if person A has a connection to person B, and person B has a connection with person C, then we expect A and B to be similar and B and C to be similar. By transitivity, we expect A and C to be similar with a probability that is significantly higher than random chance. Therefore like social contagion, latent homophily works under the mechanism of triadic closure, so there is an increased likelihood of triangles in the network. Unlike social contagions, the directionality of the link formation is symmetric, so we are as equally likely to observe a link from Person A to Person C as we are to observe a link from Person C to Person A.

Network Simulations

I developed two algorithms that simulate the latent homophily process and the social leveraging process on an input Barabasi-Albert network. For these simulations, a directed edge from node A to node B means node A possesses the potential to influence node B. In an empirical setting, the potential to influence could be represented by various proxies, such as friendship, communication, or other social constructs. Potential to influence is an asymmetric relationship, and proxy to study potential to influence can be adapted based on the attributes of the empirical network. In the homophily simulation, the likelihood of A influencing B should be the same as the likelihood of B influencing A, so the direction of edge formation between any two pairs of nodes is symmetric. On the other hand, in the social leveraging simulation, the likelihood of forming an edge from node A to node B is proportional to the number of directed paths from the former to the latter, so the direction of edge formation is asymmetric between any pair of nodes.

Latent Homophily Simulation

The latent homophily simulation algorithm looks for pairs of nodes that are not already neighbors and calculates the number of friends they have in common. The algorithm then divides this value by the geometric mean of the number of neighbors for each node to define a proper probability value. The sociological intuition for this mechanism is that the more friends node A and B have in common, the more opportunity for them to interact. However, this interaction is less significant if each node has a large number of neighbors because then any interaction between the two nodes is less meaningful, corresponding to passing each other by at large party as opposed to talking to each other during a small gathering of friends. Latent homophily implies individuals forms edges because they are similar, and similarity is a symmetric relation, this method is symmetric in both directions, so the probability of adding an edge from B to A is the same as adding an edge from A to B.

Our latent homophily simulation algorithm differs from other latent homophily simulation algorithm in that we do not pre-specify the number of communities in the network, nor does our algorithm attempt to construct latent homophily relationships from the pre-determined communities [19]. Therefore, this homophily simulation algorithm is a contribution of this thesis.

We provide the algorithm and an example of a homophily network produced by this simulation below.

Algorithm 1 Latent Homophily Simulation

```
Input: InputNetwork, NumIterations
for $i = 0$ to $NumIterations$ do
    for $NodeA$ in $InputNetwork.Nodes()$ do
        $Neighbors_A = A.getNeighbors()$
        for $NodeB$ in $InputNetwork.Nodes()$ do
            if $A \neq B$ and $B \notin Neighbors_A$ then
                                                ▷ % Find the common neighbors of the two nodes
                $CommonNeighbors = Neighbors_A \cap Neighbors_B$
                $GeometricMean = \frac{len(Common\ Neighbors)}{\sqrt{len(Neighbors_A) * len(Neighbors_B)}}$
        ▷ % Adds an edge from Node A to Node B with probability $GeometricMean$
                InputNetwork.addEdge(A,B, probability = GeometricMean)
            end if
        end for
    end for
end for
return InputNetwork
```

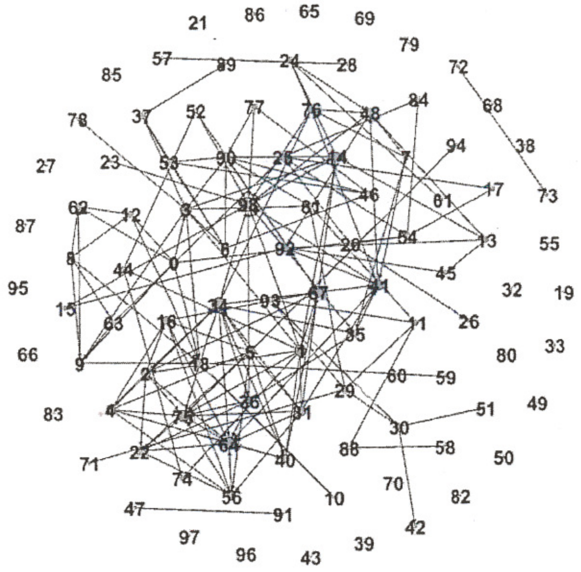

FIGURE 2: A 100-node network produced by the homophily simulation

Social Leveraging Simulation

The social leveraging simulation algorithm looks for pairs of nodes that are not already neighbors and calculates the number of paths from A to B and defines the probability of adding an edge from A to B as the number of paths divided by the total number of paths in the network to normalize the probabilities. A sociological intutition for this approach if person A has many pathways to potentially influence person C, then he has more opportunies to leverage his current connections, so as to form an edge with person C.

Because computing all the paths from A to B becomes inefficient as the network grows in size, the total number of paths from A to B were approximated by powers of the adjacency matrix. Formally, we took the adjacency matrix M and iteratively calculated the powers of this matrix up to $M^{\frac{E}{V}}$, where $\frac{E}{V}$ is the floored ratio of edges to nodes in the network. By calculating powers of the matrix, we estimated the number of paths of length $\frac{E}{V}$ from node A to node B and vice versa. This calculation is not symmetric because if there are more paths from A to B compared to the number of paths from B to A, then the probability of forming an edge from A to B is higher than the probability of forming an edge from B to A. There is no algorithm in the literature that simulates the effects of social leveraging, so this simulation algorithm is an unique contribution of this thesis.

Algorithm 2 Social Leveraging Simulation

```
Input: InputNetwork $\in \mathbb{R}^{n \times n}$, NumIterations
for $i = 0$ to $NumIterations$ do
    Initialize M $\in \mathbb{R}^{n \times n}$ to the zero matrix.
    if $InputNetwork_{ij} \neq 0$ and $i \neq j$ then
        $M_{ij}$ = Number of paths from $i$ to $j$.
    end if
    M = $\frac{M}{||M||}$                    ▷ This normalizes the matrix M
    Pick an entry $(i, j)$ of $M$ proportional to its value, and add an edge between $i$ and $j$ in the InputNetwork.
end for
return InputNetwork
```

We provide an social leverage network produced by this algorithm below:

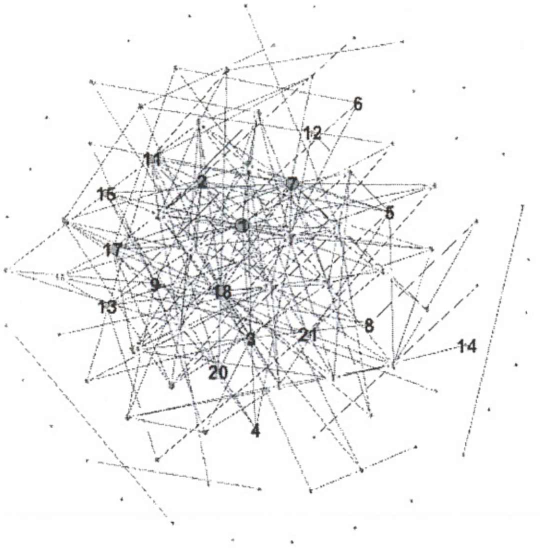

FIGURE 3: A 100-node network produced by the social leveraging simulation

Mathematical Representation of Social Networks

Graph Theoretic Representation

For the last two hundred years, networks have been represented by a square adjacency matrix $A \in \mathbb{R}^{n \times n}$ where a_{ij} denotes information about the connection between node i and node j. In an unweighted network, $a_{ij} \in \{0, 1\}$ where $a_{ij} = 1$ if node i and node j are connected, and $a_{ij} = 0$ otherwise. In a undirected network, the adjacency matrix A is symmetric. The graph-theory based representation reflects the connections between the nodes in the network, but it may not accurately capture the diffusion of information throughtout the network. For a given adjacency matrix representation, different models of information diffusion may be equally valid, and there is no further method to validate the correctness of these models.

Covariance Matrix Representation

For this thesis, we will use a different mathematical representation of networks developed by Shrivastava and Li [20]. Given an adjacency matrix A, the vector of ones $\hat{e}$, and some postive integer k, this representation uses the Krylov subspace projection of A on $\hat{e}$ up to order k, which is $\{A\hat{e}, A^2\hat{e}, \cdots, A^k\hat{e}\}$.

In a unweighted adjacency matrix, only immediate neighbors are represented. To calculate paths from non-neighboring pairs of nodes in the network, we compute powers of the adjacency matrix. For some positive integer m, A^m denotes the adjacency matrix A raised to the m^{th} power, and for this matrix, A^m_{ij} is the number of paths from node I to node J of length m.

Algorithm 3 Covariance Matrix Representation

```
Input: $A \in \mathbb{R}^{n \times n}$: Adjacency Matrix, $k$: Number of power iterations
  $x^0 = \hat{e} \in \mathbb{R}^{n \times 1}$
  Initialize Matrix $M \in \mathbb{R}^{k \times n}$
  for $t = 1$ to $k$ do
      $M_{(:),(t)} = n \times \frac{Ax^{t-1}}{||Ax^{t-1}||_1}$        ▷ Each column of $M_{(:),(t)}$ is an $n \times 1$ vector
      $x^t = M_{(:),(t)}$
  end for
  $\mu = \hat{e} \in \mathbb{R}^{k \times 1}$
  $C^A = \frac{1}{n}\sum_{i=1}^{n}(M_{(i),(:)} - \mu)(M_{(i),(:)} - \mu)^T$        ▷ $M_{(i),(:)}$ is a $k \times 1$ vector
  return $C^A \in \mathbb{R}^{k \times k}$
```

Using the provided algorithm, we generate a symmetric $k \times k$ Covariance Matrix represention of the network. Because $A\hat{e}$ is an n vector where the j^{th} element is the number of outgoing edges for node j, the j^{th} column of the matrix M reflect the proportion of outgoing paths starting from node j, and the i^{th} row of M reflects the total number of paths of length i.

As a result, the matrix returned by this algorithm, C^A, is a covariance matrix of M, reflecting the number of outgoing paths up to length k. For our model, we assume the edges in the network represent influence, so the outgoing paths represent diffusion of influence in our model. The covariance matrix C^A is also a symmetric positive-definite, meaning all of the eigenvalues of this matrix are positive. Previously, we were unable to define a notion of distance between networks of differing sizes, but with this covariance matrix representation, we can compare the resulting covariance matrix, C^A and C^B for networks A and B respectively, directly.

To demonstrate how this algorithm works, let's examine the following Barabasi-Albert network, denoted as Network A. If we run it through the covariance represen-

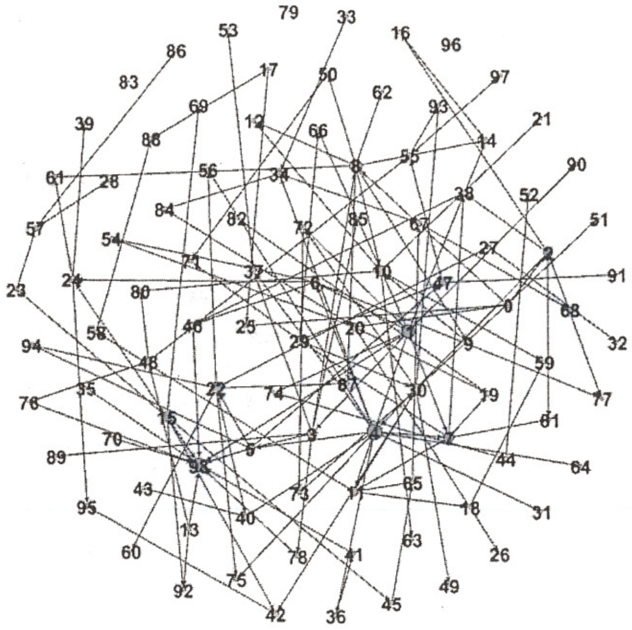

FIGURE 4: Network A: 100-node Barabasi-Albert network

tation algorithm with $k = 5$, we get back the following 5 covariance matrix, denoted by C^A. As stated above, we see the covariance matrix is symmetric and performing an eigendecompositon yields all positive eigenvalues.

$$C^A = \begin{pmatrix} 0.865 & 0.632 & 0.817 & 0.699 & 0.812 \\ 0.632 & 0.707 & 0.761 & 0.785 & 0.798 \\ 0.816 & 0.761 & 0.960 & 0.887 & 0.998 \\ 0.699 & 0.785 & 0.886 & 0.910 & 0.944 \\ 0.812 & 0.798 & 0.998 & 0.944 & 1.054 \end{pmatrix} \qquad (1)$$

Now, let's consider the following network generated by running the homophily mechanism on Barabasi-Albert network A, and denoted this network as Network B. Running it through the covariance representation algorithm with $k = 5$ yields the covariance matrix C^B.

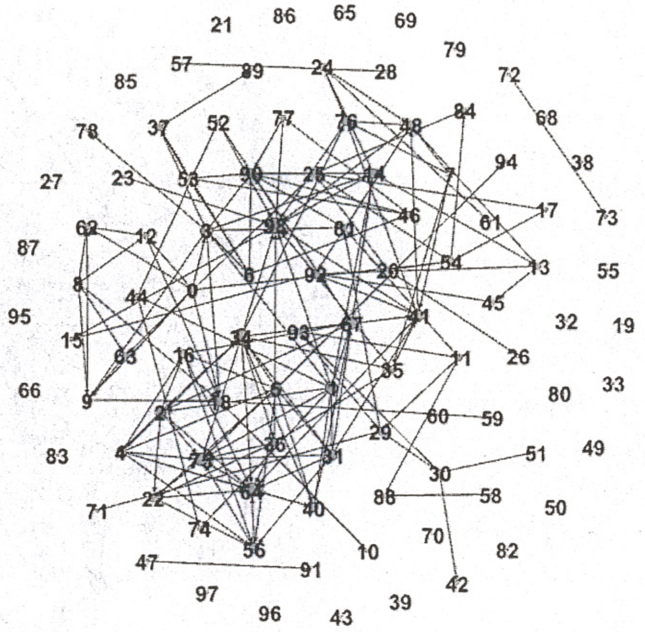

FIGURE 5: Network B: 100-node Latent Homophily network

$$C^B = \begin{pmatrix} 2.259 & 2.272 & 2.319 & 2.339 & 2.352 \\ 2.272 & 2.473 & 2.541 & 2.575 & 2.592 \\ 2.319 & 2.541 & 2.645 & 2.683 & 2.705 \\ 2.339 & 2.575 & 2.683 & 2.727 & 2.751 \\ 2.352 & 2.592 & 2.705 & 2.750 & 2.774 \end{pmatrix} \qquad (2)$$

From visual inspection, we see that in network B, the nodes are more densely connected in smaller component. As a result, the amount of information that can diffuse in this component is greater than that of the diffusion in network A. Therefore, we observe the values of C^B are greater than their corresponding values in C^A. This provides initial belief that this representation may be useful for classifying network generated by different mechanism. In addition, we see that values of k as small as 1 may be able to distinguish between the different network types.

Classification of Social Networks using Support Vector Machines

To perform classification on the social networks, we constructed the adjacency matrix for each network and transformed it into the covariance matrix representation up to some order k. We flattened this matrix into a k^2-dimensional vector, so it can be used a training example for a support vector machine. Each flattened matrix is associated with a label: 0 if the network is Barbarasi-Albert network, 1 if the network was produced by the latent homophily mechanism, and 2 if the network was produced by the social leveraging mechanism.

For classification testing, we chose to use Support Vector Machines, a machine learning algorithm used for binary classification that derives a hyperplane to maximize the margin between clusters of labeled data. Formally, this algorithm finds the hyperplane with the following properties:

$$f(x_i) = \begin{cases} w^T x_i + b = 1, & \text{if } y_i = 1 \\ w^T x_i + b = -1, & \text{if } y_i = -1 \end{cases} \tag{3}$$

where x_i, y_i are the features and the label of i^{th} training datapoint that has the smallest $||w||$ value, corresponding to the largest margin. Intuitively, for binary classification, this means if the training point has a positive label, it falls on one side of the hyperplane. If the training point has a negative label, it falls on the other side of the hyperplane. Furthermore, the hyperplane with the largest margin means the magnitude of separation between positive and negative datapoints is maximized, so no points fall within this margin, and we are more confident about our labels.

However, this formulation of Support Vector Machines works only when the data is linearly separable. Therefore, we used a soft-margin Support Vector Machine to classify the data points. Unlike the previous formulation, the soft-maring Support Vector Machine allows slack for the data points to fall within the margin.

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{4}$$

subject to the constraints

$$\xi \geq 0 \tag{5}$$

$$(w \cdot x_i + b) y_i - (1 - \xi_i) \geq 0 \tag{6}$$

where C is a cost penalty parameter that allows us to control how much importance we place on data points falling within the margin or misclassifications, and

ξ_i is the amount of slack we allow for datapoint x_i to satisfy the $(w \cdot x_i + b)y_i - (1 - \xi_i) \geq 0$ constraint. The minimization of $w^T w$ allows the classifier to learn the hyperplane with the largest margin, so the overall optimization problem becomes a trade-off between learning the hyperplane with the largest margin and learning the hyperplane with the least amount of slack.

To generalize this binary classification task to multiple classes, we found the hyperplane corresponding to each label and aggregated these hyperplanes to classify the data. The Support Vector Machine use a symmetric, positive semi-definite kernel function to define a notion of distance between the datapoints. In addition, a cost penalty C is used to penalize the Support Vector Machine if data points fall within the margin [21].

Because support vector machines can perform only binary classification, the experiment phase is done in one vs. other testing; ie. Latent Homophily Network vs Not Latent Homophily Network, Social Leveraging Network vs Not Social Leveraging Network.

Because these networks are generated entirely by a single process, we wanted to see how our classifier performed when the networks are a mixture of latent homophily and social leverage. We did not develop a network simulation algorithm that simulates mixtures of latent homophily and social leveraging, so we created these mixed networks by taking convex combinations of our pure networks. The resulting mixed network was then assigned the label of the pure network with the highest coefficient in the convex combination.

Results

To test our methodology, we simulated 600 social leverage networks, 600 homophily networks, and constructed a synthetic dataset consisting of these 1800 networks (600 social leverage, 600 latent homophily, and 600 base Barbarasi-Albert graphs). We partitioned this synthetic dataset to generate a training and testing dataset. These simulated networks ranged in size from 1000 nodes to 2000 nodes.

Support Vector Machine Classification

The Support Vector Machine was created with a *linear kernel*, meaning the notion of distance between two datapoints is defined by their inner product in Euclidean Space, and cost penalty $C = 1$. This soft-margin classifier finds hyperplane that best separates the training data but allows for misclassification. The classifier achieved 98% accuracy on the test dataset for both the Latent Homophily Network vs Not Homophily and the Social Leveraging network vs Not Social Leveraging network classification task. We applied *Stratified 2-Fold Cross Validation*, a statistical methodology to partition the dataset into training and testing subsets such that the data is evenly represented between the two groups, to generate the training and testing dataset, so the high accuracy cannot be attributed to bad representation of the labels in the testing phase [22]. Because a linear kernel yielded 98 % accuracy, this suggests there may be a mathematical difference in the covariance matrices produced by these two processes: Latent Homophily and Social Leveraging.

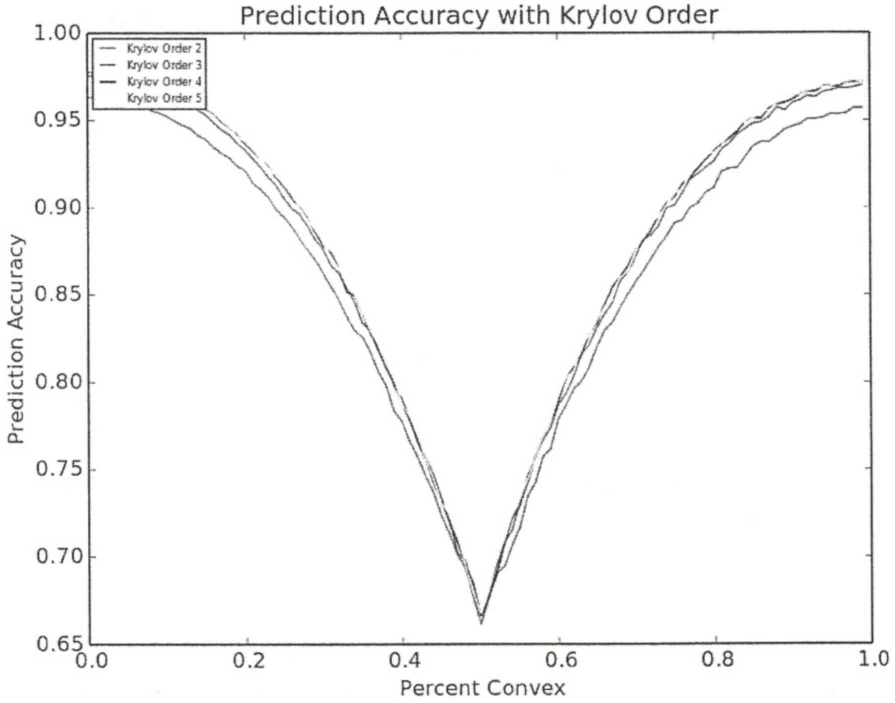

FIGURE 6: The accuracy of the classifier as the level of convexity changed

We tested our classifier on convex combinations of our homophily networks and our social leverage networks. For a given $\theta \in (0, 1)$, 400 convex combinations were created by randomly sampling with replacement 400 social leverage networks and 400 latent homophily networks. For a sampled latent homophily network L and a social leveraging network S, a convex combination C was generated by $C := \theta S + (1 - \theta)L$. In addition, the label of the convex network C was defined by $\text{label}(C) := \theta \ \text{label}(S) +$

$(1-\theta)$ label(L). We did not consider Barabasi-Albert networks in the convex combinations. The homophily networks and the social leverage networks were generated from Barbarasi-Albert networks, so a convex combination would represent a network where the latent homophily process or the social leverage process was less prominent. Furthermore, including Barabasi-Albert networks would introduce this problem of convex combinations of unordered labels, rendering classification using the labels impossible.

We see a noticable drop in the classifier accuracy as the convexity level approached $\theta = 0.5$. For different Krylov subspace projections of length k, the figure above shows the trade-off between the accuracy of the classifier and the convexity of the combination (the value of θ). Krylov subspace projections above length 5 were not computed due to lack of memory issues on our hardware. We observe a tandem increase in the classifier accuracy and the order of the Krylov subspace, suggesting that as the covariance matrix provides more information about longer paths of diffusion, the network classification problem becomes easier for the support vector machine. If the clusters are separated, then this decrease in accuracy can be explained because the convex combination pulls the two clusters together, so the two clusters overlap more as $\theta \to 0.5$, therefore making it harder for the classifier to distinguish between the two clusters.

We observe that as the value of k increases, the classifier can discriminate between different networks with higher accuracy. However, the effect on classifier accuracy by increasing k above 4 diminishes according to our results. This suggests that having more information about the pathways of influence diffusion yields a more accurate decision boundary. To test whether this new mathematical representation is better than the standard adjacency matrix, we tested the Support Vector machine trained on the covariance matrix representation vs Support Vector machines trained on the adjacency matrix representation. Because adjacency matrices are not comparable when they differ in size, both classifiers were trained on networks consisting of 1000 nodes and networks consisting of 1500 nodes. The results are summarized below.

Table 1 Classifier performance using different network representations

	Covariance Matrix (K = 5)	Adjacency Matrix
1000 nodes Networks (Convexity = 0.00)	0.983	0.331
1000 nodes Networks (Convexity = 0.55)	0.668	0.014
1500 nodes Networks (Convexity = 0.00)	0.977	0.283
1500 nodes Networks (Convexity = 0.55)	0.651	0.000

Network Motifs

To identify the motifs in the networks, we used the FANMOD application . In a manner similar to a Permutation Test, the algorithm enumerates all possible subgraphs, and then randomly adds and removes edges while simultaneously preserving bidirectional edges to calculate p-values to detect of motifs of size m,[23]. Using the FANMOD network motif detection software, we identified six network motifs that are over-represented in the homophily networks. The six motifs are presented below: All of these motifs are closed motifs, so this is consistent with out hypothesis that the latent homophily mechanism increases the representation of closed 3-node motifs.

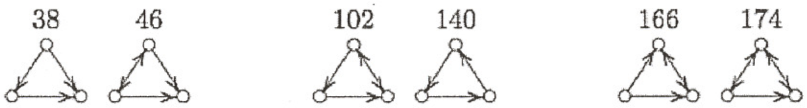

FIGURE 7: The six motifs discovered in latent homophily networks

We found three motifs that are over-represented in the social leveraging networks. These three motifs are motif 38, motif 46, and motif 166.

We found none of the three-node motif types to be over-expressed in the Barnabasi-Albert networks. We tested at the $\alpha = 0.01$ significance level, using the Bonferroni Correction to account for multiple hypothesis testing. We provide the results for 1000 node networks below.

Table 2 The frequency and p-value of each of the six network motifs in 1000-node Leverage networks

Motif ID	Leverage Freq	Leverage P-value	BA network Freq	BA network P-value
Motif 6	34.13%	1	11.498%	0.166
Motif 12	24.533%	1	20.475%	0.119
Motif 14	1.9987%	1	20.755%	0.776
Motif 36	34.490%	1	12.751%	0.199
Motif 38	2.1885%	0	0.078816%	0.872
Motif 46	0.1265%	0	0.035029%	0.27
Motif 78	0.050601%	1	10.649%	0.981
Motif 102	0.042167%	0.025	0.052544%	0.339
Motif 140	0.016867%	0.0945	0.017515%	0.763
Motif 164	2.3023%	1	23.601%	0.726
Motif 166	0.18132%	0	0.035029%	0.352
Motif 174	0.016867%	0	0.035029%	0.785
Motif 238	0.0042167%	0	0.017515%	0

Table 3 The frequency and p-value of each of the six network motifs in 1000-node homophily networks

Motif ID	Homophily Freq	Homophily P-value	BA network Freq	BA network P-value
Motif 6	22.082%	1	10.306%	0.526
Motif 12	42.742%	1	19.815%	0.002
Motif 14	4.7679%	1	23.654%	0.412
Motif 36	18.481%	1	9.4377%	0.484
Motif 38	4.8163%	0	0.079643%	0.744
Motif 46	0.443565%	0	0.023893%	0.605
Motif 78	0.28559%	1	13.42%	0.914
Motif 102	0.5615%	0	0.023893%	0.941
Motif 140	0.79868%	0	0.0%	1
Motif 164	4.4871%	1	23.168%	0.433
Motif 166	0.37756%	0	0.023893%	0.592
Motif 174	0.15974%	0	0.039822%	0.861
Motif 238	0.0048405%	0	0.0079643%	0.004

We found motif 12 to be statisically significant in only two of the Barabasi-Albert networks, so we believe these over-representations may be due to random fluctuations. In addition, we found motif 238 to be statistically significant in half of the

Barabasi-Albert networks, but because the overall frequency of this particular motif is low, so this result can be attributed to random fluctuations.

The motifs 38, 46, and 166 are statistically significant in all of our social leveraging networks, but we also found motifs 178, 238, and 78 to be significant in some of the leveraging networks. Motifs 78, 178, and 238 are closely related in that adding one edge to motif 78 produces motif 178 and adding two edges to motif 78 produces motif 238. These motifs occur at such a low frequency, so these results can be explained by random fluctuations.

Motifs 38, 46, and 166 are closed motifs that do not induce a cycle in the sub-graph, that is to say it is impossible to explore all three nodes in the motif following the directions of the arrows. On the other hand, motifs 102, 140 and 174 do induce a cycle in the sub-graph. From this result, we may distinguish latent homophily networks from social leverage networks by the presence of closed, cyclical three-node motifs. The latent homophily mechanism favors all closed three-node motifs, but the social leveraging mechansim is more restrictive by favoring only acyclical closed three-node motifs. The presence of closed triangles is an indicator of latent homophily, so this result shows both mechanisms operate according to triadic closure.

Conclusion

We examined two different approaches to studying the evolution of network structures under different mechansisms. The first approach uses network motifs as an observable local feature of the network to identify whether a network evolved according to latent homophily or social leveraging. We found only acyclic closed three-node motifs are over-expressed in the social leverage networks whereas all closed three-node motifs are over-expressed in the latent homophily networks. The over-expression of close motifs show both mechanisms operate under triadic closure, One possible avenue for future work involves designing statistical tests that use the over-expression of network motifs to determine if the evolution of two social networks occurs due to different processes acting on the networks.

In the second approach, we used a new mathematical representation of social networks based on the Krylov Subspace projection of the adjacency matrices to represent the network as a covariance matrix. For some integer k, networks of different sizes can all be represented as a $k \times k$ covariance matrix. The covariance matrix representation allows us to compare networks of different sizes metrically. When these covariance matrices are used as features for a Support Vector Machine classifier, these classifiers achieve near-perfect accuracy on the classification test. The classifier shows robustness in its accuracy when convex combinations of networks are introduced into the dataset. Support Vector Machines trained using covariance matrices as features greatly outperform Support Vector Machines trained using adjacency matrices as features. We have shown the covariance matrix representation provides more insight compared to the adjacency matrix in distinguishing between different network evolution processes.

We have laid down the theoretical framework for this methodology to study the evolution of networks. One possible follow-up to this work is constructing a similarity metric for network evolution processes. Since all of this work was doing using simulated data, future work would apply these methodolgies to an empirical dataset. Because our model assumes the directed edges show potential to influence, one of the main challenges in applying this to a real-world dataset is understanding how to determine the presence, magnitude, and direction of potential to influence.

Appendix

The full results of the FANMOD algorithm are included below. If an individual cell value has an asterisk, then this motif frequency is statistically significant. If the motif name is bolded, then all values in that row are statistically significant.

Table 4 The frequency each of the network motifs in homophily networks

Motif ID	1100 Nodes	1200 Nodes	1300 Nodes	1400 Nodes	1500 Nodes	1600 Nodes	1700 Nodes	1800 Nodes	1900 Nodes
Motif 6	21.989%	22.053%	22.834%	21.845%	22.366%	23.251%	22.495%	23.158%	23.54%
Motif 12	42.518%	42.987%	43.957%	43.925%	42.927%	43.637%	44.366%	43.224%	44.761%
Motif 14	4.9689%	4.267%	3.916%	4.2269%	4.792%	4.2968%	3.8962%	4.424%	3.5245%
Motif 36	17.689%	19.217%	19.194%	19.134%	18.351%	18.724%	19.08%	18.246%	19.39%
Motif 38	5.335%	5.0244%	4.6396%	4.5293%	4.8366%	4.4178%	4.2567%	4.5949%	3.8772%
Motif 46	0.47597%	0.39632%	0.27551%	0.26726%	0.39376%	0.30864%	0.35184%	0.4132%	0.26124%
Motif 78	0.2406%	0.21577%	0.13225%	0.18286%	0.2786%	0.18155%	0.19899%	0.23922%	0.14804%
Motif 102	0.65903%	0.50641%	0.39674%	0.57671%	0.58692%	0.43573%	0.49027%	0.49397	0.39839%
Motif 140	1.0722%	0.90273%	0.74205%	0.78419%	0.81724%	0.68688%	0.79019%	0.70523%	0.6335%
Motif 164	4.3674%	3.972%	3.5523%	3.9772%	4.0565%	3.6492%	3.5991%	3.9642%	3.1283%
Motif 166	0.47597%	0.34788%	0.26817%	0.3833%	0.40862%	0.34193%	0.36049%	0.38213%	0.26559%
Motif 174	0.19876%	0.10568%	0.080817%	0.15824%	0.18202%	0.069596%	0.10671%	0.14291%	0.071841
Motif 238	*0.010461%	*0.0044035%	*0.007347%	*0.01055%	*0.0037147%	0.0%	*0.00865%	*0.012427%	0.0%

Table 5 The frequency each of the network motifs in leverage networks

	1100 Nodes	1200 Nodes	1300 Nodes	1400 Nodes	1500 Nodes	1600 Nodes	1700 Nodes	1800 Nodes	1900 Nodes
Motif 6	32.104%	39.945%	34.042%	34.573%	34.601%	34.53%	35.337%	37.442%	34.428%
Motif 12	26.831%	23.758%	25.013%	23.746%	22.064%	27.461%	22.628%	27.093%	24.608%
Motif 14	1.0464%	0.84743%	1.1936%	1.4607%	1.6337%	1.8131%	0.94829%	1.5931%	1.4151%
Motif 36	35.73%	32.397%	35.962%	36.118%	37.859%	32.475%	37.358%	30.012%	35.966%
Motif 38	2.2366%	1.8298%	2.3773%	2.2885%	2.0704%	1.8483%	2.1463%	2.3417%	1.9259%
Motif 46	0.070007%	0.044996%	0.1061%	0.11301%	0.11293%	0.086895%	0.072272%	0.066905%	0.095196%
Motif 78	*0.018423%	*0.018748%	*0.016578%	0.036728%	*0.035134%	*0.037576%	0.01095%	*0.039782%	0.029291%
Motif 102	0.018423%	0.0074993%	0.026525%	0.014126%	0.010038%	0.018788%	0.0065702%	0.016274%	0.010984%
Motif 140	0.055269%	0.018748%	*0.046419%	0.019777%	0.0050191%	0.044622%	0.021901%	0.04159%	0.020138%
Motif 164	1.7833%	1.0462%	1.1406%	1.5624%	1.5133%	1.5946%	1.3535%	1.2821%	1.4279%
Motif 166	0.10685%	0.086242%	0.069629%	0.056505%	0.092853%	0.086895%	0.1095%	0.063289%	0.067736%
Motif 174	0.0%	0.0%	*0.0066313%	*0.011301%	0.0025095%	0.0023485%	*0.0087602%	*0.0090413%	0.0054921%
Motif 238	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Table 6 The frequency each of the network motifs in the Barabasi-Albert networks

	1100 Nodes	1200 Nodes	1300 Nodes	1400 Nodes	1500 Nodes	1600 Nodes	1700 Nodes	1800 Nodes	1900 Nodes
Motif 6	10.244%	10.839%	9.7098%	11.286%	10.009%	10.813%	11.127%	10.949%	11.114%
Motif 12	22.306%	22.41%	20.905%	22.584%	21.756%	22.338%	22.344%	*22.954%	23.13%
Motif 14	22.039%	21.538%	22.502%	22.689%	21.937%	22.349%	22.154%	21.276%	21.802%
Motif 36	10.994%	11.645%	10.889%	11.027%	11.7%	11.468%	11.043%	11.609%	11.286%
Motif 38	0.058392%	0.10344%	0.056162%	0.037044%	0.046636%	0.022788%	0.022434%	0.01583%	0.044118%
Motif 46	0.016683%	0.022166%	0.024961%	0.037044%	0.023318%	0.017091%	0.011217%	0.026384%	0.016043%
Motif 78	11.395%	11.009%	11.85%	10.817%	11.245%	10.967%	11.122%	10.464%	10.56%
Motif 102	0.066733%	0.066499%	0.049922%	0.043218%	0.029148%	0.051273%	0.022434%	0.026384%	0.0080215%
Motif 140	0.0%	0.014778%	0.01248%	0.006174%	0.0058295%	0.005697%	0.016826%	0.0052768%	0.0080215%
Motif 164	22.806%	22.255%	23.906%	21.418%	23.167%	21.922%	22.064%	22.59%	21.975%
Motif 166	0.025025%	0.0073888%	0.037441%	0.018522%	0.029148%	0.017091%	0.011217%	0.021107%	0.016043%
Motif 174	0.05005%	0.088666%	0.043682%	0.024696%	0.046636%	0.017091%	0.061694%	0.063321%	0.032086%
Motif 238	0.0%	0.0%	*0.01248%	*0.012348%	0.0058295%	*0.011394%	0.0%	0.0%	*0.0080215%

Bibliography

[1] S. Milgram and J.Travers. An experimental study of the small world problem. *Sociometry*, 1969.

[2] M.S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 1973.

[3] T. Kuhn, M. Perc, and D. Helbing. Inheritance patterns in citation networks reveal scientific memes. *arXiv*, 2014.

[4] Alex Pentland. *Social Physics: How Good Ideas Spread - The Lessons from a New Science*. The Penguish Press HC, 2014.

[5] P. Erdos and A. Renyi. On the evolution of random graphs. *Publications of the Mathematica Institute of the Hungarian Academy of Science*, 1960.

[6] D.J. Watts and S.H Strogatz. Collection dynamics of small-world networks. *Nature*, 1989.

[7] A.L. Barabasi and R. Albert. Statistical mechanics of complex networks. *Review of Modern Physics*, January 2002.

[8] P. Borgnat and E. Fleury. Evolving networks. *Nature*, 2007.

[9] G. Palla, A.L Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 2007.

[10] S. Feizi, D. Marbach, M. Medard, and M. Kellis. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature Biotechnology*, 2013.

[11] David Choi. Testing for coordination and peer influence in network data. Machine Learning and Social Science Seminar, 2013.

[12] A. Masoudi-Nejad, F. Schreiber, and MK Razaghi. Building blocks of biological networks: A review of major network motif discovery algorithms. *IET System Biology*, 2012.

[13] D. Conway. Modeling network evolution using graph motifs. *arXiv*, May 2011.

[14] M.E.J. Newman. The structure and function of complex networks. *Physical Review*, 2003.

[15] R. Milo, SS Shen-Orr, S. Itzkovitz, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 2002.

[16] C.R. Shalizi and A.C Thomas. Homophily and contagion are generally confounded in observational social network studies. *arXiv: Sociological Methods and Research*, April 2010.

[17] N. Friedkin. A structural theory of social influence. *Cambridge University Press*, 1998.

[18] P. Toulis and E. Kao. Estimation of causal peer influence effects. *International Journal of Machine Learning*, 2013.

[19] P. Krivitsky, M. Handcock, A. Raftery, and P. Hoff. Representing degree distribution, clustering, and homophily in social networks with latent cluster random effect models. *Social Networks*, 2009.

[20] A. Shrivastava and P. Li. A new mathematical space for social networks. *Neural Informations Processing Systems Foundation*, December 2013.

[21] A.Smola and B. Scholkopf. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.

[22] T. Hastie, J. Friedman, and R. Tibshirani. *The Elements of Statistical Learning.* Springer Series in Statistics, 2009.

[23] S. Wernick and F. Rasche. Fanmod: A tool for fast network motif detection. *Bioinformatics*, February 2006.