2014

Modeling Tolerance in Dynamic Social Networks

Amanda Luby
College of Saint Benedict/Saint John's University

Modeling Tolerance in Dynamic Social Networks

An Honors Thesis

College of St. Benedict/St. John's University

In Partial Fulfillment of the Requirements
For Distinction in the Department of Mathematics
And the Department of Computer Science

Amanda Luby
Advisor: Dr. Yu Zhang

April 15th, 2014

Approval Page

Approved By:

Dr. Yu Zhang, Advisor
Associate Professor of Computer Science

Dr. Kris Nairn, Reader
Associate Professor of Mathematics

Dr. Phil Byrne, Reader
Professor of Mathematics

Dr. Bob Hesse, Department Chair
Chair, Department of Mathematics

Dr. Imad Rahal, Department Chair
Chair, Department of Computer Science

Dr. Tony Cunningham
Director, Honors Thesis Program

Abstract

The study of social networks has become increasingly important in recent years. Multi-agent systems research has proven to be an effective way of representing both static and dynamic social networks in order to model and analyze many different situations. Previous implementations of multi-agent systems have observed a phenomenon called tolerance between agents through simulation studies, which is defined as an agent maintaining an unrewarding connection. This concept has also arisen in the social sciences through the study of networks. We aim to bridge this gap between simulation studies in multi-agent systems and real-world observations. This project explores how local interactions of autonomous agents in a network relate to the development of tolerance. We have developed a new model for multi-agent system interactions based on these observations. We also claim that tolerance is directly observable in real dynamic social networks, and the parameters that govern tolerance of a system can be estimated using a Hidden Markov Model.

Contents

1 Introduction

1.1 Motivation

A social network is made up of a set of actors as well as the set of ties between them. Examining the structure of the whole network, as well as individual patterns that arise, offers valuable insight into many different social applications. These include

- Studies of Communication, which focuses on the study of the transfer of information. This can include in-person communication, such as the spread of a rumor; or public-forum communication, such as information conveyed on a blog. (Fleming 2011 [12]; Minsheng, Xinjun and Guessoum 2013 [30]; and Zhou 2013 [46])

- Community development, including both geographical and online communities. Of particular interest is developing tools to analyze the development of social media networks such as facebook, twitter, and wordpress. (Lachapelle 2011 [21]; Zhoua, Dingb and Fininc 2013 [46])

- Diffusion of innovations, or the spread of ideas throughout a community. This can include finding the 'opinion leaders', or the individuals who are especially influential in the spread of an idea, as well as modeling the spread of an innovation through an entire organization. Recent studies into diffusion have also looked at how diffusion interacts with network structure. (Stattner, Collard and Vidot 2013 [41])

- Health care analysis, including epidemological studies and studies of health care organizations and systems. (Levy and Pescosolido 2002 [23]; Christakis and Fowler 2013 [4])

- Language and linguistics, including how different languages evolve through social interaction. In an increasingly globalized world, this is of particular interest in studying the decline of native dialects as well as language maintenance and shift in multi-lingual communities. (Milroy 2008 [29])

- Social Capital, or the resources available to individuals through their social interactions. For instance, social capital allows certain people to access opportunities such as job openings. It has also been shown that there is a correlation between measured social capital and reported quality of life. (Valenzuela, Park, and Kee 2009 [42])

As social networks can be used to analyze anything from social networking websites to interactions between animals, being able to effectively study them

has become increasingly important in recent years. (Pinter-Wollman, Noa and Hobson 2013 [34])

We are particularly interested in dynamic social networks, which are networks where connections are being continuously made and broken. Dynamic social networks can include infinitely growing networks, such as the World Wide Web(Zhoua, Dingb, and Fininc 2013 [46]; Schwanda and Bazarovab 2014 [37]); or networks in which the number of actors remains the same but the connections between them change over time, such as a trust network within a class (Genicot 2011 [14]). Additionally, dynamic social networks may be actor-oriented, in which the actors control both their outgoing ties and behavior, or the network may be governed by the structure as a whole. (Snijderes 2005 [39])

We have approached our study of dynamic social networks through both Multi-Agent systems and Markov Chains. A Markov Chain is a set of random variables which change at each time-step according to certain transition probabilities. A Markov Chain is similar to a Multi-Agent System in the sense that it evolves over a series of time-steps, but different in that it is rooted in probability theory. This concept is applicable to the study of dynamic social networks because it allows them to change over time, rather than remaining static. Additionally, Markov chains may be extended to include multiple states within each chain as well as multiple chains combined into one model. This makes them particularly useful for modeling attributes about networks. For instance, having multiple states representing a single chain can be used to model whether or not a person has an infection based on their outward symptoms. Additionally, having multiple chains within one system could model the health of one person based on the health of the other people that are also represented by Markov chains. (Dong 2011 [8])

Multi-Agent systems is an area of research under the large umbrella of artificial intelligence. Agents perceive their environment and perform actions based on those perceptions. A Multi-Agent system is an environment in which multiple agents interact with each other, as well as the environment. These agents must be rational, or choose the action that maximizes their success out of all possible actions. Usually, this measure of success can be determined through game theory using simulations, or some sort of function to determine payout from data. (Wooldridge 2009 [43])

Previous research into MultiAgent systems has observed a phenomenon, named tolerance, through simulations. (Wu 2010 [44]) Tolerance is defined as agent's willingness to maintain an unrewarding connection. In other words, during these simulations, agents would remain connected even though they should have broken the connection according to the decision-making rule being tested.

Intuitively, this agrees with what we observe in the real world. If two people have a relationship, they are unlikely to completely break off from one another just because of one disagreement or bad decision by one person. Instead, they usually remain in the relationship but may distance themselves and possibly break it off at a later time, depending on the behavior (Milardo 1986 [28], Ojanen and Sijtsema 2010 [32]).

This research attempts to bridge the gap between tolerance observed in MultiAgent system simulations and sociological ideas about maintaining relationships. We do this through introducing a new decision-making rule for a multi-agent system, as well as using a Hidden Markov Model to essentially measure observed tolerance in a real-world dynamic social network. By using a Hidden Markov Model to study a dynamic social network, we are able to estimate the parameters that govern the system. We can then incorporate these parameters into the multi-agent system to best model the network.

1.2 Background

In multi-agent systems, an agent is defined as an entity that is able to perceive its environment and proceed to act upon that environment. Agents are assumed to be rational, meaning that they will choose the action that will cause them to be the most successful. A multi-agent system contains a number of agents that each have their own goals and are able to interact with one-another.

Through our study of dynamic social networks, an agent will represent a person in the network, and the environment is the status of the network at a given time. (i.e. the state of connections) The agents are given an opportunity to interact with each of their connections at each time step, and are able to maintain, break, or create a connection based on those interactions. This choice is the action that they are able to make on the environment. Agents learn from their interactions with other agents and can change their behavior based on what they observe. As optimal strategies are determined by agents, a structure starts to form. This allows us to study how the structure of the dynamic social network evolves.

Previous research into social networks has delved into interaction rules, or how agents should behave in a given situation. These studies originated in static networks, in which only the actions changed from step to step but connections were never broken (Shoham and Tennenholtz 1997 [38], Delgado 2002 [7]). However, they have also been extended to dynamic social networks, in which agents not only decide on a given action, but are also given the choice to make or break a connection given the actions of the other agents in the system (Leezer and Zhang 2009 [22], Wu and Zhang 2010 [44]).

In simulation studies, game theory is utilized to determine payout to model observations about the real world. Most of the research in this area has been restricted to simulation studies, or synthetic data, as relevant, real data was unavailable. However, due to recent advances in cell-phone technology, we have access to a dataset that is very applicable to social network studies. This dataset includes cell phone statistics from 80 residents in a single dormitory over a nine month period and includes information about proximity, location, phone calls, and text messages. Additionally, the participants of the study completed periodic surveys regarding friendship with other participants of the study. (Dong 2011 [8])

When applying the theoretical models developed to actual data, we run into problems with how to deal with temporal data. A possible solution is to think of the state of the network as a Markov chain. A Markov chain is a series of random variables observed at multiple time-steps with the conditional independence property. That is, the state variable at a given time step is only dependent on the state variable at the previous time step. They also rely on a transition probability which gives the probability that a state will change from one time-step to another. If the transition probabilities are chosen correctly, the Markov chain will eventually converge.

An extension to the Markov chain, that is more applicable to modeling an attribute of a dynamic social network, is the Hidden Markov Model. This involves an underlying Markov chain, which is the 'hidden' variable, but also has observation variables. So in addition to the transition probabilities associated with the underlying Markov chain, the model also needs emission probabilities that determine what is observed in the chain of observable variables.

A Hidden Markov Model seems like a natural solution to our problem. It is intuitive to think of the status of a relationship as a random variable and also to think of the relationship being able to change at each time-step, much like a Markov chain. If we treat each possible relationship as a Markov chain, we can then represent the network as the set of all these possible relationships. We can also utilize an observable trait of phone calls as the observable variable. By implementing the use of a hidden Markov model, it is possible to determine the parameters that govern the status of a relationship.

As previously discussed, the concept of tolerance in relationships has arisen in recent research. That is, even though the relationship is unrewarding and should be broken according to the rule, the agents delay breaking the relationship for a few time steps (Wu and Zhang 2010 [44]).

The concept of homophily has been well-defined and studied in the social sciences. This is the tendency of individuals to associate and bond with others that are similar to themselves. In the context of social networks, it

has been observed that ties between non-homophilic individuals dissolve at a higher rate than ties between homophilic individuals (McPherson, Smith-Lovin, and Cook 2011 [27]). This implies that individuals have a certain level of tolerance in which they will maintain a connection with qualities or actions that they disagree with, and will break that connection if the level of tolerance is breached. (Genicot 2011 [14]) We observe the same trend in our experiment, with certain connections being maintained for long periods of time while other connections are created and broken in a small period of time.

1.3 Research Goal

To our knowledge, we are the first group to observe and study tolerance in dynamic social networks through the use of a multi-agent system. Although a similar phenomenon has been observed in studies in sociology and psychology, a computational model with a statistical analysis has not been proposed. The aim of this research is to bridge the gap between simulation studies in multi-agent systems research and sociological observations.

We attempt to model this phenomenon through the use of a Multi-Agent system as well as a Markov Chain. Both of these approaches give us a stochastic method to analyze a network. Through the use of a Multi-agent system, we are able to model tolerance directly and preserve the actor-based decision making that is found in the real world. By implementing a hidden markov model, we are able to refine the system through formal parameter estimation as well as a statistical analysis.

This project explores how local interactions of autonomous agents in a network relate to collective behaviors. The collective behavior of interest is tolerance. We claim that tolerance is directly observable in real dynamic social networks, and the parameters that govern tolerance of a system can be estimated using a Hidden Markov Model.

1.4 Approach

As previously mentioned, the concept of tolerance was first observed in the context of a multi-agent system through simulation. For this reason, we have defined a new decision-making rule that allows for an agent to make the explicit choice to 'tolerate' an unrewarding connection with another agent. This allows us to study tolerance more effectively and efficiently, as it is built into the system and has become easier to measure. The multi-agent system has been utilized as a simulation study, attempting to obtain results similar

to the real dataset being studied. We then use these results to estimate the underlying parameters that determine the decision-making strategies used in the real dataset.

Additionally, we have employed a Hidden Markov Model to estimate the transition and emission probabilities for the dataset chosen. We then performed a statistical analysis of the estimated probabilities to determine if there was a significant difference in tolerance between friends and non-friends. The significant difference in observed tolerance provides quantitative evidence of tolerance observed in the real-world.

2 Related Work

2.1 Tolerance and Friendship in Social Sciences

As previously discussed, homophily has been extensively studied and defined in the social sciences. Homophily can be summarized as 'birds of a feather flock together', or that individuals with much in common have a higher rate of interaction than individuals without common traits. This is especially important in the study of social networks, as connections are more likely to occur between homophilic individuals. There is also a tendency for relationships to disband when homophily was not observed (McPherson, Smith-Lovin, and Cook 2001 [27]).

This tendency to disband brings up the sense of tolerance, although not explicitly named. It seems that individuals may be more tolerant of homophilic members of a network, and less tolerant with non-homophilic members. However, a social network with only ties between completely homophilic individuals is virtually impossible in the real world. Thus there exists some sort of equilibrium for making and maintaining connections with non-homophilic individuals. This implies that, most of the time, a given individual will have a certain amount of tolerance towards any other individual in the network. (Currarini, Jackson, and Pin 2007 [5])

It has also been proposed that individuals can be characterized by a level of tolerance for behaviors that differ from their ideal, and that when tolerance levels must differ in societies for an equilibrium to occur. We must also make a distinction between tolerance for other's types versus tolerance for other's behaviors. Types could include characteristics such as religion, ethnicity, sexual orientation, age, or social status; while behaviors can be expressed regardless of underlying type. (Genicot 2011 [14])

There has been additional evidence of tolerance through studies of the formation of friendships and social structure. It has been shown that people are unlikely to break a relationship based on one disagreement or a bad decision made. Instead, they maintain the relationship but may distance themselves and possibly break the connection at a later point in time (Ojanen and Sijtsema 2010 [32]).

When discussing tolerance and friendship in the context of studying a social network, we must also take the structure of the network into consideration. One consequence of this is the 'friends of friends' idea. In a longitudinal friendship network analysis, it was found that friendships tend to be transitive, which leads to the formation of triplets. Transitive friendship triplets link each individual closely to the other members of the triplet, and eventually to the friends of the others. It was also shown that when belong-

ing to a triplet, a person is unlikely to change a relationship in this triplet, even when the relationship is unrewarding, contributing to the evidence for observed tolerance (Ojanen 2010 [32]; Jackson and Rogers 2007 [16]).

2.2 Dynamic Social Networks

A Social Network can be thought of as a group of agents with connections between them. In a dynamic social network, these connections can be changed over time. Much of the research into these networks has focused on the different ways that they can develop. Generally, they have been classified into three different categories. (Toivonen 2009 [35])

1. Dynamic Network Evolution Models (NEM): These models have a fixed number of nodes, and create and break connections between the nodes based on triadic closure and global connections (Davidson 2002 [6]; Marsili 2004 [26]; Kumpula 2007 [20]; Snijders, Lomi, and Torlo 2013 [40]; Krivitsky and Handcock 2014 [19])

2. Growing Network Evolution Models: These models have similar rules as dynamic NEM's, but continuously add nodes instead of maintaining a fixed number. In addition, they do not break connections like dynamic NEM's. (Ivanova and Iordanov 2012 [15]; O'Malley and Onella 2014 [33])

3. Nodal Attribute Models (NAM): Unlike the previous two models, a NAM does not rely on network structure and creates connections based explicitly on the attributes of two given nodes. (Boguna 2004 [2]; Myunghwan and Jure 2011 [31]; Fosdick and Hoff 2013 [13])

Although all of these rules have provided insight into how social networks change over time, it has been hard to find examples of theory that has been applied to real-world situations. One of the problems that arises is how to use various types of social network data to determine the structure of the network and how it changes.

2.3 Interaction Rules

In MultiAgent systems, agents will encounter each other, and we need to define rules for interaction for when these encounters happen. This is especially important in dynamic social networks, as the agents need to know when they should make or break a connection. However, it is also important in static social networks when we want to study the structure of the network or the

strength of connections.

Existing Rules:

- Highest Current Reward (HCR): Agents keep track of the payoff received from the last play of each strategy. Then at each time step, they select the action that previously earned them the largest reward. Agents change their strategy in the event that it earns them a payoff that is less than the previous reward earned from another strategy. (Shoham and Tennenholtz 1997 [38])

- Generalized Simple Majority (GSM): Agents will change to an alternative strategy if they have observed more instances of it on other agents than their present action. (Delgado 2002 [7])

- Highest Rewarding Neighborhood (HRN): An agent will maintain a relationship if and only if the average reward earned from that relationship is no less than a specified percentage of the average reward earned from every relationship. (Leezer and Zhang 2009 [22])

- Highest Weighted Reward (HWR): An agent will maintain a relationship if and only if the weighted average reward earned from that relationship is no less than a specified percentage of the weighted average reward earned from every relationship. (Wu and Zhang 2010 [44]) (Where recent events are weighted more than past events)

- Pay-and-Call (PaC): This method was developed to qualify interactions between agents based on mobile phone communication. Agents want to maximize their payoffs, which depend on the friendliness between agents, the length of communication, as well as cost of initializing and maintaining the communication. (Joseph 2013 [17])

2.4 Markov Models

Markov Models have been utilized to model a variety of problems. Although much of recent research in the area has focused on extensions of a simple markov chain, the underlying theory of the model has proven to be incredibly useful. Hidden Markov Models (HMM) have been especially useful to researchers in the areas of speech recognition and EEG classification (Zhong and Ghosh 2002 [45], Krishnan and Fernandez 2013 [18]). They have also been used for applications as far reaching as monitoring volcano activity(Cassisi 2013 [3])Here, the underlying Markov Chain is associated

with observed variables, and a series of transition and emission probabilities determine the dynamics of the system.

An increasingly widely used extension of the HMM is the Coupled Hidden Markov Model (CHMM). Like an HMM, a CHMM contains an underlying Markov Chain. However, a CHMM is made up of more than one HMM, whose states depend not only on their previous states, but on all the previous states of all the HMM's that are contained in the CHMM. CHMM's have been shown to be useful in the areas of multi-channel EEG classification, complex human action recognition, traffic modeling, and biosignal analysis. (Zhong and Ghosh 2002 [45])

Due to the sequential nature of Markov Models, they have often been proposed to model various properties of dynamic social networks. However, most of these proposals have been extended implementations of the Coupled Hidden Markov Model, including the Graph-Coupled HMM, where the latent state of the Markov chain is dependent only on the previous states of the chains that it has a connection to. (Dong, Pentland, and Heller 2012 [9]). These models have been applied to modeling the spread of an infection, but also have the capability to model opinion changing in a community, culture formation, vocabulary imitation, and dynamics of fads, rumors, emotions, opinions, etc.

2.5 Social Evolution Data

The Social Evolution Dataset provided by the MIT Human Dynamics Lab has been used for many different types of research. It was originally used as an argument that mobile phone data can enhance studies on social aspects of communities (Madan 2012 [25]). It was shown that mobile phone data supported the traditional methods of studying factors such as individual symptoms, long-term health outcomes, and diffusion of opinions in a community.

Other studies modeled infection throughout the community using both MultiAgent systems (Dong 2012 [9]) and graph-coupled hidden markov models (Dong 2012- SBP [10]). They were able to successfully predict how infection is spread using the Social Evolution dataset. Other health-related applications include modeling obesity and healthy eating using a social network, which showed that there was a correlation between relationships and eating and fitness habits. (Madan 2010 [24])

Researchers have also shown that relationships and behavior co-evolve in this dataset. This means that students in the dormitory changed their behavior as their relationships changed; but also that their relationships changed based on their behavior. For instance, they showed that individuals were

more likely to have a friendship if they exercised the same number of times per week. They also showed that friendships can determine how frequently an individual visits different places. (Dong 2011 [8])

3 Highest Tolerated Reward

The Highest Tolerated Reward (HTR) Neighborhood Evaluation Rule builds off of previously proposed neighbor evaluation rules - namely the Highest Weighted Reward (HWR) rule. The HTR rule consists of an evaluation method as well as a connection-forming mechanism. If an agent's average reward is between a range of thresholds, they will search for new connections without breaking any. If they drop below this range, they will begin breaking connections until their average reward increases.

Definition 1. The *Highest Tolerated Reward* rule states that an agent will maintain a relationship if and only if the time-discounted average reward from that relationship is above a certain maintaining threshold. An agent will seek new relationships when their average reward is below a certain seeking threshold. When the agent is between the maintaining threshold and the seeking threshold, we call the agent *tolerant*.

3.1 Neighborhood Evaluation

In order to define the HTR rule, we make use of previously defined equations and procedures to measure rewards from each neighbor (Wu and Zhang 2010 [44]) . However, since the nature of previous papers has been focused on coding a simulation, we have redefined some equations and changed terminology to maintain mathematical rigor.

Definition 2. The Payoff Function, $p_{ij}(t)$, is defined as the payoff between agents i and j at time t.

Definition 3. The Reward Function, $r_{ij}(t)$ is the cumulative reward between agents i and j at time t.

Definition 4. The time-discount factor, w, is a proportion between 0.5 and 1.0 and allows recent rewards to carry a heavier weight than earlier rewards.

For each neighbor j that agent i has, the cumulative reward $r_{ij}(t)$ is the time-discounted reward from agent j.

$$r_{ij}(t) = r_{ij}(t-1) * w + p_{ij}(t) \tag{1}$$

Note that

$$r_{ij}(t) = \sum_{k=1}^{t} p(k) * w^{t-k} \tag{2}$$

Proof.

$$
\begin{aligned}
r_{ij}(t) &= r_{ij}(t-1) * w + p_{ij}(t) \\
&= [r_{ij}(t-2) * w + p_{ij}(t-1)] * w + p_{ij}(t) \\
&= r_{ij}(t-2) * w^2 + p_{ij}(t-1) * w + p_{ij}(t) \\
&= [r_{ij}(t-3) * w + p_{ij}(t-2)] * w^2 + p_{ij}(t-1) * w + p_{ij}(t) \\
&= r_{ij}(t-3) * w^3 + p_{ij}(t-2) * w^2 + p_{ij}(t-1) * w + p_{ij}(t) \\
&\vdots \\
&= p_{ij}(1) * w^{t-1} + p_{ij}(2) * w^{t-2} + \ldots + p_{ij}(t-1) * w + p_{ij}(t) \\
&= \sum_{k=1}^{t} p_{ij}(k) * w^{t-k}
\end{aligned}
$$

□

Definition 5. The Reward Average Function, $R_i(t)$ is the cumulative weighted reward average agent i has received from all connections at time t.

$$R_i(t) = R_i(t-1) * w + \frac{\sum_{j=1}^{n} p_{ij}(t)}{n} \tag{3}$$

Note that instead of defining the Reward Average Function recursively, we can equivalently use

$$R_i(t) = \frac{1}{n} \sum_{j=1}^{n} r_{ij}(t) \tag{4}$$

Proof.

$$
\begin{aligned}
R_i(t) &= R_i(t-1) * w + \frac{\sum_{j=1}^{n} p_{ij}(t)}{n} \\
&= [R_i(t-2) * w + \frac{\sum_{j=1}^{n} p_{ij}(t-1)}{n}] * w + \frac{\sum_{j=1}^{n} p_{ij}(t)}{n} \\
&= R_i(t-2) * w^2 + \frac{\sum_{j=1}^{n} p_{ij}(t-1)}{n} * w + \frac{\sum_{j=1}^{n} p_{ij}(t)}{n} \\
&\vdots \\
&= \frac{\sum_{j=1}^{n} p_{ij}(1)}{n} * w^{t-1} + \frac{\sum_{j=1}^{n} p_{ij}(2)}{n} * w^{t-2} + \ldots + \frac{\sum_{j=1}^{n} p_{ij}(t-1)}{n} * w + \frac{\sum_{j=1}^{n} p_{ij}(t)}{n} \\
&= \frac{1}{n}[p_{i1}(1) * w^{t-1} + \ldots + p_{in}(1) * w^{t-1} + \ldots + p_{i1}(t) + \ldots + p_{in}(t)] \\
&= \frac{1}{n}[p_{i1}(1) * w^{t-1} + \ldots + p_{i1}(t) + \ldots + p_{in}(1) * w^{t-1} + \ldots + p_{in}(t)] \\
&= \frac{1}{n}[r_{i1}(t) + r_{i2}(t) + \ldots + r_{ij}(t) + \ldots + r_{in}(t)] \\
&= \frac{1}{n}\sum_{j=1}^{n} r_{ij}(t)
\end{aligned}
$$

□

According to the HTR rule, an agent has an upper threshold, θ, as well as a lower threshold, ϕ, and calculates the cumulative reward received from each neighbor as well as their weighted reward average from all connections. Then, the agent evaluates neighbors according to the two thresholds. Finally, the agent decides on a course of action based on the following rule:

$$
\begin{cases}
\frac{r_{ij}}{R_i} > \theta & \text{Keep the Connection} \\
\alpha < \frac{r_{ij}}{R_i} < \theta & \text{Make a new Connection} \\
\frac{r_{ij}}{R_i} < \alpha & \text{Break the Connection and Replace it}
\end{cases} \tag{5}
$$

3.2 Connection Formation

In addition to adding another threshold for evaluating the neighborhood, this rule also involves a mechanism for choosing new connections. We use a modified version of a dynamic network evolution model (Marsili 2004 [26]). The connection formation rule depends solely on the network structure rather

than attributes of the nodes in the network.

$$\begin{cases} \text{Connect i to a random node with probability } \gamma \\ \text{Connect } i \text{ to a friend of a friend by uniformly random search with probability } \delta \\ \text{Where } \gamma + \delta = 1 \end{cases} \tag{6}$$

By using this connection formation algorithm, it will naturally arise that the network will contain more triangles than in a network where new connections are chosen randomly (Newman 2003).

Definition 6. Three agents in a network, a,b,c form a triangle if there is a connection from a to b, b to c, and c to a.

We can then measure how many 'friends of friends' connections exist in a social network using the number of triangles in a network.

Definition 7. The *Clustering Coefficient (C)* of a given network is the mean probability that two vertices that are network neighbors of the same other vertex will themselves be neighbors.

$$C = \frac{6 * \text{Number of Triangles}}{\text{Number of Paths of Length 2}} \tag{7}$$

Note that the Clustering Coefficient is used interchangeably with transitivity of a network.

4 Tolerance

An observed consequence of using a model of this design is agents remaining in a relationship for a number of time steps even though the relationship is unrewarding (Wu and Zhang 2010 [44]). We call this phenomenon tolerance.

Definition 8. *Tolerance* is an agent's willingness to maintain an unrewarding connection. An agent that chooses to maintain an unrewarding connection for n turns is *n-tolerant*.

Suppose two agents (i and j) have been good neighbors for k turns but agent j has changed to an unfair strategy. This means that up until that turn, the agents have been cooperating but now agent j is no longer cooperating. Thus the relationship has become unrewarding to agent i. We want to determine how long agent i will tolerate the relationship with agent j before breaking the connection. We assume that after the switch made by agent j, neither agent changes behavior until the connection is broken. For simplicity, we define $p_{ij}(k+1) = Q$, i.e. Q represents the reward gained at the first non-cooperative turn. For any c-turns after the k-cooperative turns, we have that

$$r_{ij}(k+c) = w^c \sum_{m=1}^{k} p_{ij}(m) w^{k-m} + Q \times (\frac{1-w^c}{1-w}) \tag{8}$$

provided that $w \neq 1$. where

- $r_{ij}(k+c)$ is the reward obtained at the current turn by agent i from agent j
- w is the time-discount factor
- k is the number of cooperative turns that have already happened
- c is the number of turns since agent j has switched to an uncooperative strategy
- p_{ij} is the payout at a given turn by agent i from agent j
- Q is the reward gained from a non-cooperative turn

Then equation 8 gives a definition for the total cumulative reward after c turns of an non-cooperative partnership ($r_{ij}(k+c)$) in terms of only the individual payoffs (p_{ij}) and the time-discount factor (w). Additionally, the right-hand side of the equation is split up into two parts - the first being the reward from the cooperative period of the relationship, and the second

being the reward from the non-cooperative period of the connection. We assume that p_{ij} may vary while the agents are cooperating. We make this assumption because certain behaviors may be less rewarding than others, but still be considered a cooperative behavior. We also make the assumption that the non-cooperative reward, Q, remains constant until the relationship is broken.

Proof.

$$\begin{aligned}
r_{ij}(k+1) &= r_{ij}(k)w + Q \\
r_{ij}(k+2) &= (r_{ij}(k)w + Q)w + Q \\
&= r_{ij}(k)w^2 + Qw + Q \\
r_{ij}(k+3) &= r_{ij}(k)w^3 + Qw^2 + Qw + Q \\
&\vdots \\
r_{ij}(k+c) &= r_{ij}(k)w^c + Q\sum_{m=0}^{c-1} w^m \\
&= \sum_{m=1}^{k} p_{ij}(m)w^{n-k}w^c + Q\sum_{m=0}^{c-1} w^m \\
&= w^c \sum_{m=1}^{k} p_{ij}(m)w^{k-m} + Q(\frac{1-w^c}{1-w})
\end{aligned}$$

□

By defining the total cumulative reward (r_{ij}) in terms of previous payoffs (p_{ij}) rather than recursively, it is possible to determine the n-tolerance of an agent by using c from equation (8). Once a connection has been broken, the n-tolerance for that pair is c.

5 Markov Model

An alternative approach to modeling dynamic social networks, outside of a multi-agent system, is to think of the status of the network as a random variable that changes over a series of timesteps. In this scenario, using a Markov Chain to model the state of the network seems like an intuitive approach to take.

A Markov Chain is a series of random variables observed at multiple time steps with the conditional independence property. That is, the observation at a given time step is only dependent on the observation at the previous time step. They also rely on a transition probability which gives the probability that a state will change from one time step to another. If the transition probabilities are chosen correctly, the Markov chain will eventually converge to the actual distribution of the Markov model.

While looking at real-world data, it is often found that the state of interest is latent and not directly observable. Hence the concept of a Hidden-Markov Model (HMM) was developed. Rather than a single state with a single transition probability, there is a random variable state as well as observable variables that are assumed to be dependent on the states. In addition to the state transition probabilities, there are also emission probabilities associated with each output.

A Markov Chain seems like a natural solution to modeling a dynamic social network. It's intuitive to think of the status of each member of a social network as a random variable and also to think of the network changing in a series of time steps, like a Markov Chain.

In the following, we introduce the theory behind the Markov Model as well as how it will be applied to the Social Evolution dataset in order to obtain evidence of tolerance. We will be using two of the Social Evolution datasets, one with observed phone calls over the 298-day time period, and one with periodic survey data over the same time period. The data will be discussed in more detail in the following section.

5.1 Markov Chains

The underlying principle of Markov Models essentially depends on the product rule for probability when looking at the joint probability for a sequence of observations (Bishop 2006 [1]). That is,

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = \prod_{n=1}^{N} (p(\mathbf{x}_n | \mathbf{x}_1, ..., \mathbf{x}_{N-1})) \tag{9}$$

Where $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n$ are matrices of size $M \times M$, where M is the number of nodes in the network and each (i, j) entry represents whether there is a connection between nodes i and j at $t = 1, 2, ..., n, ..., N$ Where N is the total number of timesteps. By making the assumption that each of the conditional distributions $p(\mathbf{x}_i|\mathbf{x}_1, ..., \mathbf{x}_{i-1})$ is independent of all previous observations except the most recent, we obtain a first-order Markov chain.

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1})) \tag{10}$$

This process can be expanded to include more observations in predicting the next value through higher-order Markov chains. For instance, if we allow the values to depend on the previous two observations, we obtain a second-order Markov chain.

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1) \prod_{n=3}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_{n-2}) \tag{11}$$

Similarly, we can extend to an K^{th} order Markov chain, which each observation depending on the previous K values of that observation. However, this comes at the cost of more parameters in the model, which becomes impractical for large K.

By modeling the network in this way, we are able to retain a very large amount of information about the network. However, when we approach the tolerance problem, a Markov chain is no longer sufficient. Tolerance happens when nodes maintain a connection even when the relationship is unrewarding. To model this using a Markov chain, we would essentially need two different chains to represent two different networks. We have the underlying 'friendship' network, which is the network of interest, but we also have a network made up of phone call data. To measure tolerance, we would want to examine the relationship between the two networks, which a single Markov chain cannot handle.

5.2 Hidden Markov Models

To address the issue, we turn to an extension of a Markov chain, the Hidden Markov Model. In an HMM, in addition to the observed variables, latent variables are also introduced. These latent variables are the ones of interest, and each latent state has a probability associated with observing each state.(Bishop 2006 [1]) Thus we have a set of observations of a variable $\mathbf{Y}$ in

addition to the latent variables $\mathbf{X}$. Here, we assume that the state of x_n depends on x_{n-1}. In the tolerance case, $\mathbf{X}$ would be the underlying 'friendship state' of the network, while $\mathbf{Y}$ would be the observed network constructed from phone calls. Thus we have that x_n represents if there is an underlying friendship between two given nodes at time n, while y_n represents whether or not a phone call was observed between the same two nodes at time n.

To find the conditional probability distribution $p(x_n|x_{n-1})$, we must first define the transition probabilities, which we represent in a matrix, $\mathbf{A}$. The (i, j) entry of A represents the probability that an observation in state i will transition to state j at any given time. So we have that $\mathbf{A}_{(0,0)}$ is the probability that non-friends remain non-friends, $\mathbf{A}_{(0,1)}$ is the probability that non-friends become friends, $\mathbf{A}_{(1,0)}$ is the probability that friends break the friendship, and $\mathbf{A}_{(1,1)}$ is the probability that friends remain friends. Since they are probabilities, we have that $0 \leq \mathbf{A}_{ij} \leq 1$ with the sum of each row being equal to one. However, since x_1 does not have a previous time step to depend on, we must define a vector of probabilities, $\boldsymbol{\pi}$, to determine the initial state of x_1.

In addition to the transition probabilities, we must also define the conditional probabilities for the observed variables, $p(\mathbf{Y}|\mathbf{X})$. In order to do this, we define $\boldsymbol{\phi}$, a set of parameters governing the conditional probability distribution. We then define $p(y_n|x_n, \boldsymbol{\phi})$ as the *emission probabilities*. However, since y_n is an observed variable, for a given value of $\boldsymbol{\phi}$ we have a vector of 2 numbers corresponding to the two possible states of the binary variable x_n.

The emission probabilities can then be written in the form:

$$p(y_n|x_n, \boldsymbol{\phi}) = \prod_{k=1}^{2} p(y_n|\boldsymbol{\phi}_k)^{x_n k} \tag{12}$$

We can now give the joint probability distribution over both latent and observed variables:

$$p(\mathbf{Y},\mathbf{X}|\boldsymbol{\theta}) = p(x_1|\boldsymbol{\pi})[\prod_{n=2}^{N} p(x_n|x_{n-1}, \mathbf{A})] \prod_{m=1}^{N} p(y_m|x_m, \boldsymbol{\phi}) \tag{13}$$

Where:

- $\mathbf{Y} = \{y_1, y_2, ..., y_N\}$ are binary variables representing whether a call or not was observed for a given pair of agents over t= {1,2,...,N} days.
- $\mathbf{X} = \{x_1, x_2, ..., x_N\}$ are binary variables representing the underlying friendship between each pair of nodes over t= {1,2,...,N} days.

- $\boldsymbol{\beta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$ are the parameters that determine the model. (The initial state probability vector, the state transition probability matrix, and the emission probability vector, respectively)

Since we are already working with an established dataset, we are provided with values for $\mathbf{X}$ and $\mathbf{Y}$. The goal for implementing a HMM in this project is to estimate the parameters that are governing the system. This will allow us to better understand the relationship between phone calls and friendship, and how often connections are broken. Having these estimates will allow us to better understand observed tolerance in a system.

5.3 Expectation-Maximization Algorithm

In order to estimate the parameters of the system, we will utilize the well-established Expectation-Maximization Algorithm (EM Algorithm). This algorithm is used generally to solve maximum likelihood estimation problems without closed-form solutions, meaning the solution cannot be represented in terms of functions and mathematical operations from a generally accepted set. (Bishop 2006 [1], Roche 2012 [36])

Each iteration in the algorithm goes through two steps - the approximation step and the maximization step. What makes the algorithm different from other two-step maximum likelihood algorithms is that it is not dependent on the probability distribution of either variable. The only assumption that must be made when using the EM algorithm is that $\mathbf{X}$ is a Markov chain, which we have already defined as such.

6 Data Analysis

All following experiments were performed using the Social Evolution Dataset provided by the MIT Human Dynamics Lab[1]. The experiment involved monitoring the phone records of 84 participants over a 9-month period. The participants all lived in the same dormitory, and also participated in six surveys regarding friendship over the nine-month period. (Survey dates: 09/09/08; 10/19/08; 12/13/08; 03/05/09; 04/17/09; 05/18/09)

Before proceeding with modeling the underlying "friendship" dynamic social network using the phone call observations, it is necessary to confirm a correlation between friendship and phone calls. Previous analysis using the Social Evolution dataset has utilized proximity data, SMS records, as well as a call log to analyze phone use. For simplicity, we restrict to using only the call log. (Eagle, Pentland, and Lazer 2009 [11])The survey data collected from participants includes five different levels of relationship:

- Close Friend
- Socialize at least twice per week
- Discussed politics since the last survey
- Shared all tagged facebook photos
- Shared blog/live journal/Twitter activities

The call dataset provided the following information about every phone call recorded for every participant in the study over the time period:

- User ID
- Destination Phone Hash
- Duration
- Destination User ID (if destination user was also participant in the study)
- Time Stamp of the form Month/Day/Year Hour/Minute/Second

From the call log, Destination Phone Hash was removed, as well as any observations where the User ID or Destination User ID was unknown. From there, the following information was extracted:

[1] http://realitycommons.media.mit.edu/socialevolution.html

- Frequency: The number of times UserID called Destination UserID
- Total Duration: The total duration of phone calls between UserID and Destination User ID
- Frequency Received: The number of times Destination UserID called UserID

The survey and call data was then combined. For each UserID and Destination UserID, in addition to the Frequency, Total Duration, and Frequency Received, we also have whether or not UserID classified Destination User ID as a Close Friend, Political Discussant, etc. Then, a logistic regression was applied to each of the friendship classifiers, using Frequency, Total Duration, and Frequency Received as the predictors. The following results were obtained

Response	N Dev.	N DF	Res. Dev.	Res. DF	G	G DF	P
Blog	614.62	466	462.82	463	151.80	3	0
Facebook	616.74	466	462.16	463	154.58	3	0
Politics	627.94	466	479.97	463	147.97	3	0
Socialize	580.19	466	416.23	463	163.96	3	0
CloseFriend	647.29	466	526.33	463	120.96	3	0

Table 6.1: Logistic Regression Results. Note that N Dev. = Null Deviance; N DF = Null Degrees of Freedom; Res. Dev = Residual Deviance; Res. DF = Residual Degrees of Freedom; G = Chi-Square value from the test; G DF = Degrees of Freedom of G; and P= p-value

Here, we have performed a likelihood ratio test (G-test). The Null Deviance represents how well the response is predicted by a model with nothing but an intercept. We use this value as a chi-square value on 466 degrees of freedom. The residual deviance represents how well the response is predicted with the new variables introduced, on 463 degrees of freedom. Then $G = \text{Null Deviance} - \text{Residual Deviance} \sim \chi^2$ Thus G follows a Chi-Squared distribution with 3 degrees of freedom (for the 3 added predictors- Frequency, Total Duration, and Frequency Received). Therefore, we conclude that each of the P-values is significant, and the predictors are meaningful to the response. For simplicity, we will restrict to one of the friendship classifiers. Since G represents the difference between the null model and the new model, it makes sense to choose the response that gives the largest G value. Thus we will restrict to the Socialize Twice Per Week relationship.

7 Results

All experiments were performed in R. R is a free, open-source software environment developed for statistical computing and graphics. Since we have a strong need to use statistical methods as well as visualize results, this was a natural choice. We have made extensive use of the *HiddenMarkov*, *plyr*, and *chron* packages in the following sections.

7.1 Algorithm Settings

To use the Expectation-Maximization algorithm to estimate the parameters for the Hidden Markov Model, we must first set the parameters with initial conditions. Although they can be set randomly, we can also use prior information known about the data to optimize our results. The three parameters that need to be estimated are $\boldsymbol{\pi}$, the intial state probability vector, $\mathbf{A}$, state transition probability matrix, and $\boldsymbol{\phi}$, the emission probability vector.

Recall that:

- $\mathbf{X}$ represents the latent state of friendship between two nodes. i.e. if $X_n = 0$ the pair are not friends at day n and if $X_n = 1$ the pair are friends at day n.
- $\mathbf{Y}$ represents the observed state of phone calls between two nodes. i.e. if $Y_n = 0$ there was not a phone call between the pair on day n and if $Y_n = 1$ there was a phone call between the pair on day n
- π_0 is the probability that a given pair of nodes are not friends on the first day
- π_1 is the probability that a given pair of nodes are friends on the first day
- $\mathbf{A}_{(0,0)}$ is the probability that a pair who were not friends at time t are still not friends at time $t + 1$
- $\mathbf{A}_{(0,1)}$ is the probability that a pair who were not friends at time t become friends at time $t + 1$
- $\mathbf{A}_{(1,0)}$ is the probability that a pair who were friends at time t stop being friends at time $t + 1$
- $\mathbf{A}_{(1,1)}$ is the probability that a pair who were friends at time t continue being friends at time $t + 1$

- ϕ_0 is the probability that non-friends call eachother at a given t
- ϕ_1 is the probability that friends call eachother at a given t

To estimate $\boldsymbol{\pi}$, which determines the beginning state of friendship for each pair, we turn to the survey data. Since $\pi_0 = P(X_1 = 0)$ and $\pi_1 = P(X_1 = 1)$, we simply compute the observed frequencies of each state for the first survey and then divide by the total number of responses. We obtain

$$\boldsymbol{\pi} = (\pi_0, \pi_1) = (\frac{6181}{7056}, \frac{875}{7056}) = (0.876, 0.124)$$

We also use the survey data to estimate $\mathbf{A}$. By counting the number of times $X_i = 0$ and $X_{i+1} = 0$, $X_i = 0$ and $X_{i+1} = 1$, $X_i = 1$ and $X_{i+1} = 0$, $X_i = 1$ and $X_{i+1} = 1$, we obtain the following transition probability matrix.

$\mathbf{A} =$

	$X_{t+1} = 0$	$X_{t+1} = 1$
$X_t = 0$	0.942	0.058
$X_t = 1$	0.339	0.661

Thus we have that

$$\mathbf{A}_{(0,0)} = 0.942, \mathbf{A}_{(0,1)} = 0.058, \mathbf{A}_{(1,0)} = 0.339, \mathbf{A}_{(1,1)} = 0.661$$

Since our observed variables follow a binomial distribution,

$$\boldsymbol{\phi} = \{P(Y_i = 1|X_i = 0), P(Y_i = 1|X_i = 1)\}$$

To begin with, we'll assume that each of the emission probabilities are the same. Thus we are assuming that friends and non-friends have equal probability of making a phone call to each other. Since the algorithm will adjust the estimates based on the data, we can then see if the estimates are changed. If they are significantly changed, we will conclude that friends are more likely to make a phone call at any given time than non-friends, or potentially vice-versa. The estimates were found by computing the observed number of days with a call divided by the total number of entries.

$$\boldsymbol{\phi} = (\phi_0, \phi_1) = (.00176, .00176)$$

Each pair was treated as an individual HMM, and the resulting parameter estimates for each observation were stored, in addition to the final log-likelihood value and the number of iterations.

7.2 Parameter Estimates

The following results were obtained.

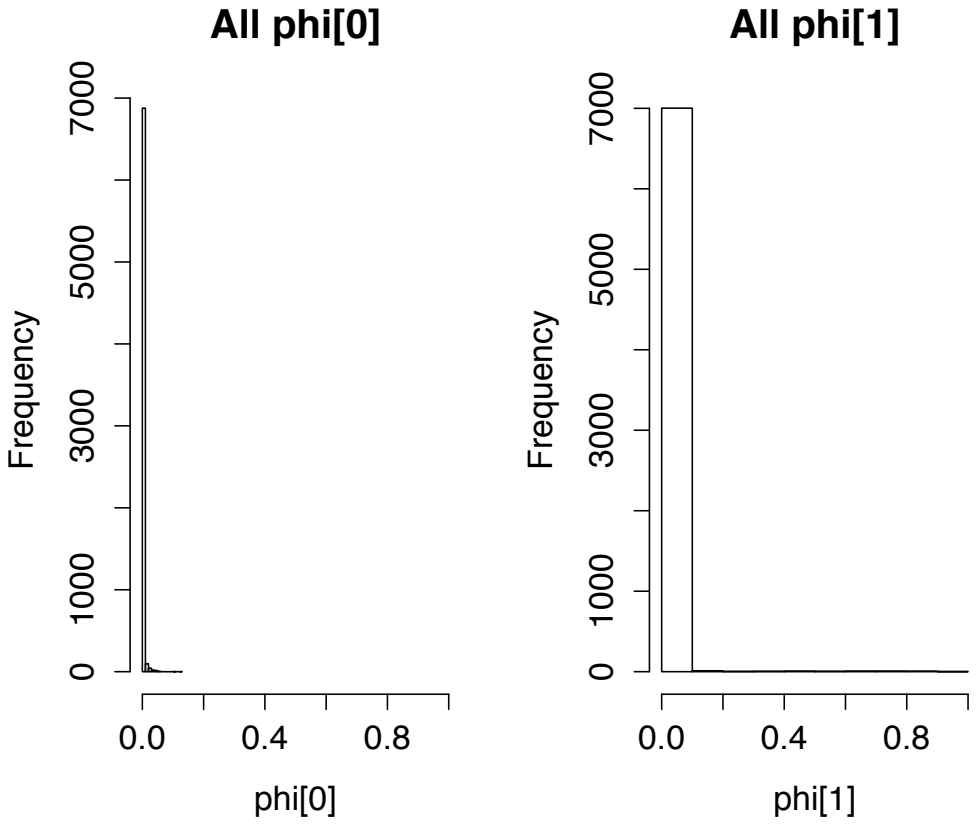

Figure 7.1: Distribution of ϕ final estimates

It is clear that the majority of HMM's produced a final estimate of $\boldsymbol{\phi}$ near zero. Upon closer inspection, for each of the 6,674 pairs that never had a phone call observed between them, the estimates of $\boldsymbol{\phi} = (0, 0)$. This should come as no surprise, since $\boldsymbol{\phi}$ is the emission probability, and if there is never a call observed, the probability of observing a call will naturally be estimated at zero.

However, due to the large size of the estimates with a value of zero, we cannot say much about the other 400 samples in the study without removing those estimates and observing.

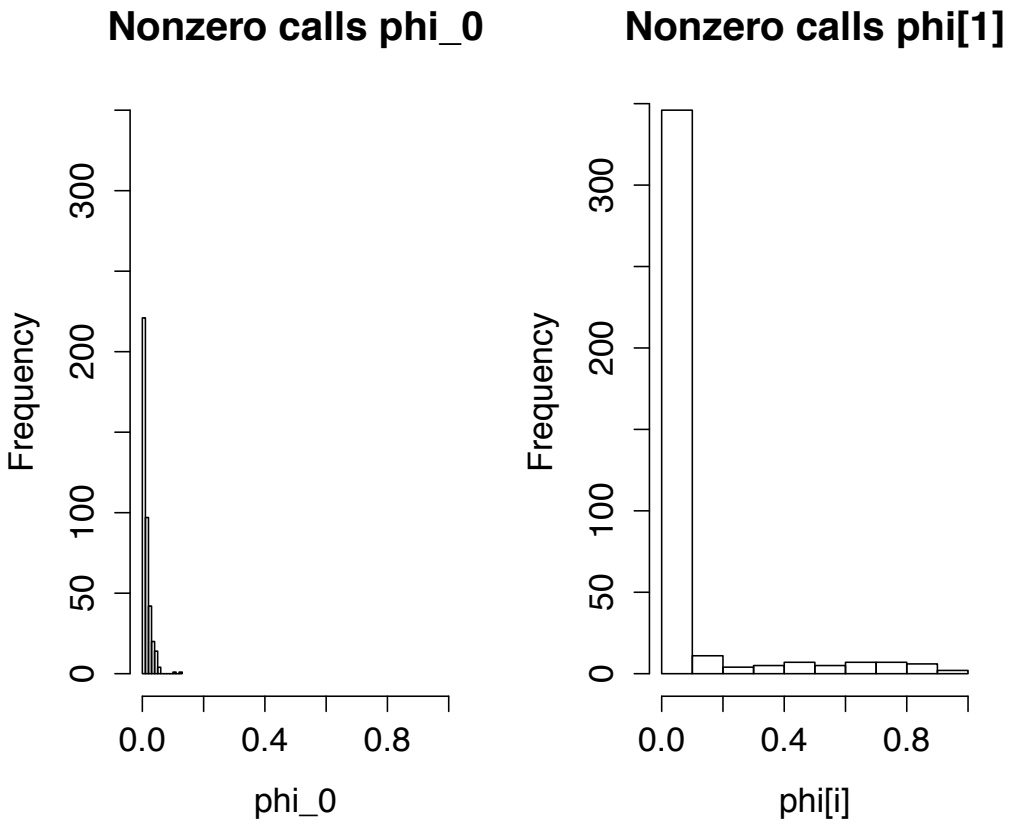

Figure 7.2: Distribution of $\boldsymbol{\phi}$ final estimates (At least 1 call observed)

Here, we see a lot more information about how the estimates for $\boldsymbol{\phi}$ change for the pairs where there is a call observed. We continue to observe that ϕ_0, although not exactly zero, remains less than 0.1 for the majority of samples. This is what would be expected, since $\phi_0 = P(Y_t = 1|X_t = 0)$, or the probability that a call is observed given the pair are not friends.

As we are focusing on tolerance, the ϕ_1 estimate is of real importance. Recall that $\phi_1 = P(Y_t = 1|X_t = 1)$, or the probability that a call is observed given the pair are friends. Since we define tolerance as maintaining an unrewarding friendship, we are interested in the probability that a call is not observed given the pair are friends. ($P(Y_t = 0|X_t = 1)$) Note that:

$$P(Y_t = 0|X_t = 1) = 1 - P(Y_t = 1|X_t = 1) = 1 - \phi_1$$

The distribution is presented below.

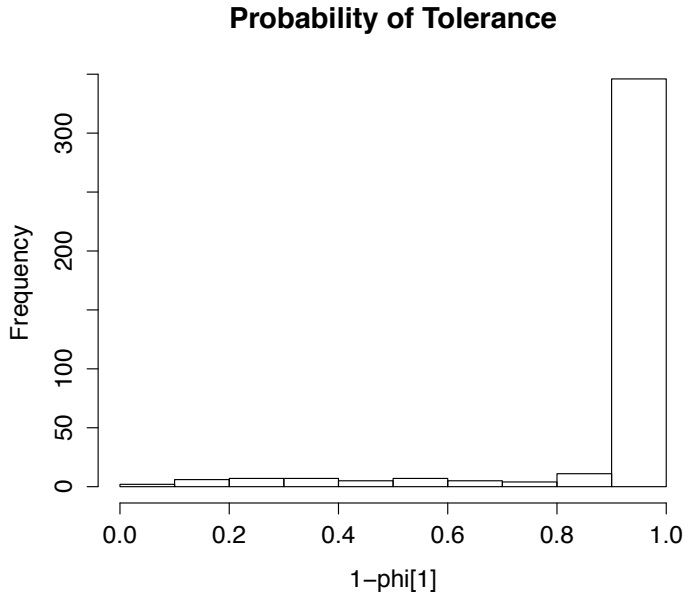

Figure 7.3: Distribution of P(Tolerance)

We see that among current friends, the probability of tolerance is concentrated at values at or near 1. In fact, 75 percent of the estimates are greater than 0.9764. The boxplot of the estimates is shown below. (Note that the dotted lines are all below the 25th percentile)

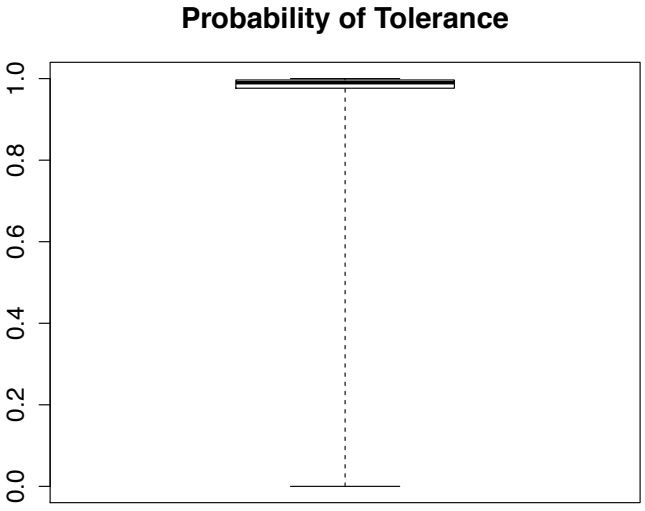

Figure 7.4: P(Tolerance) Boxplot

We now move to looking at the estimates for the transition probability matrix, **A**. The following are the estimates for all of the 7,074 samples.

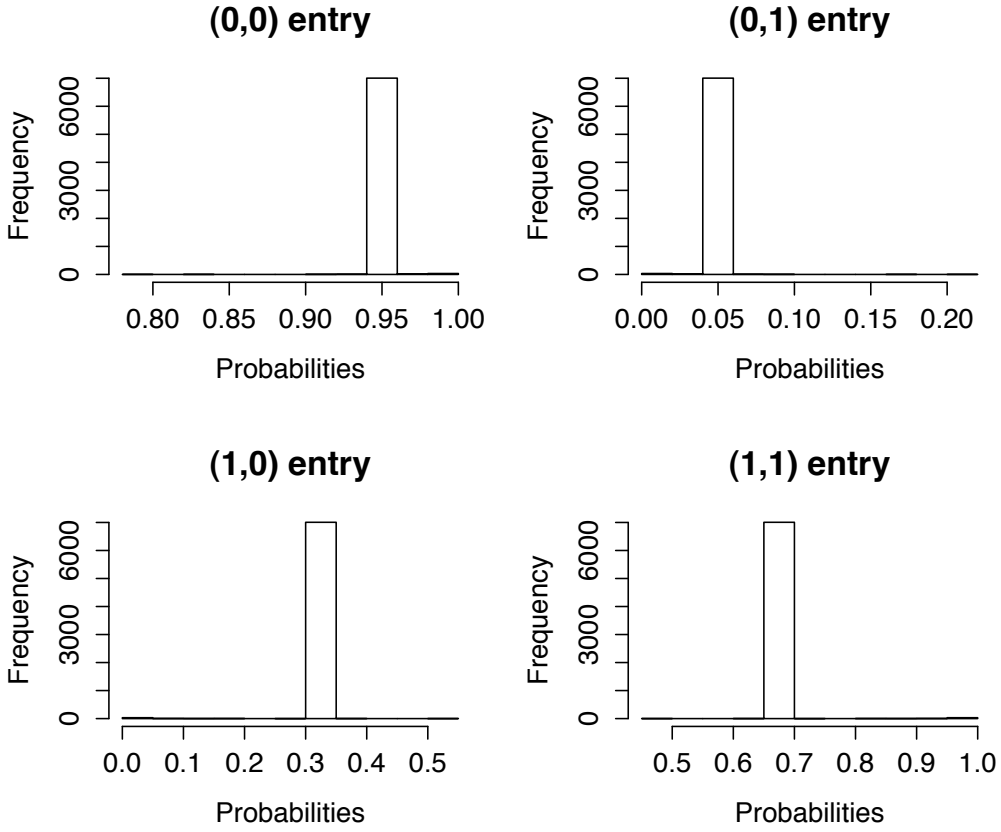

Figure 7.5: Distribution of **A** final estimates

As seen in figure 7.5, in the overwhelming majority of the HMM's, **A** did not change. This is because of the 7074 possible pairs of callers, only 400 pairs had an observed call over the 298-day time period. If there are no observed phone calls, as previously discussed, $\boldsymbol{\phi}$ naturally gets estimates of 0, and the algorithm has no information available to change the estimates of **A**.

Thus, we again want to observe only the results where the estimates had the possibility of being changed - the 400 samples with an observed phone call over the time period. These results were pulled out of the total resulting data and are shown below.

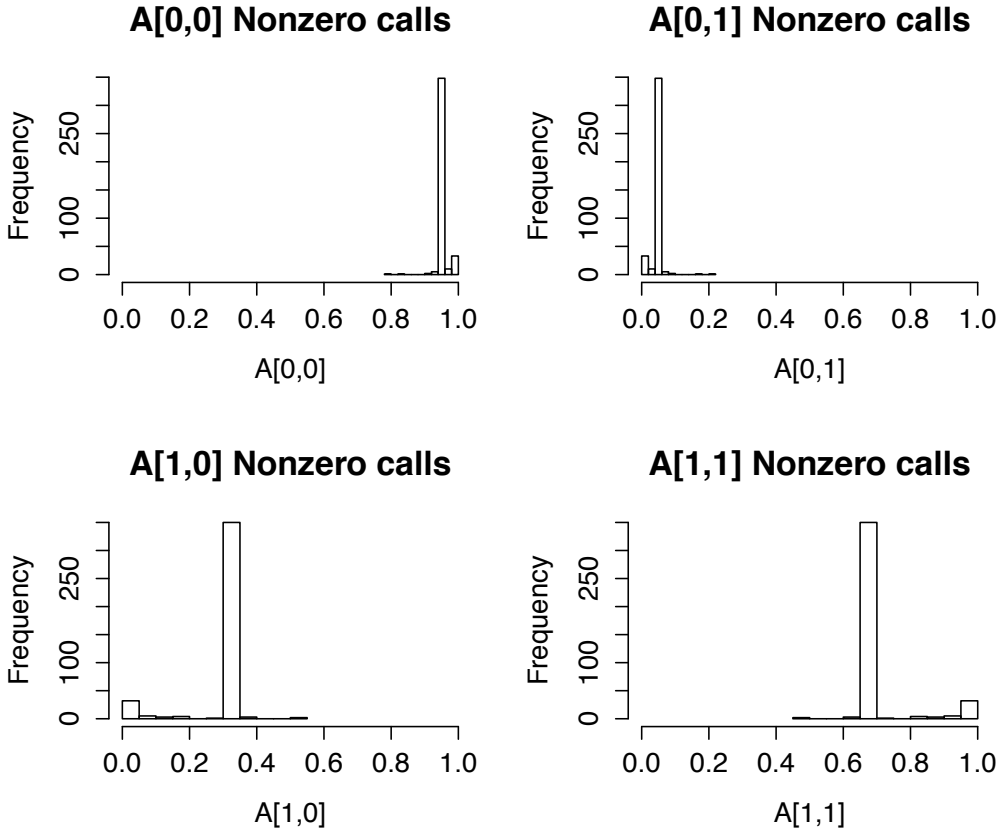

Figure 7.6: Distribution of $\mathbf{A}$ final estimates (At least 1 call observed)

We observe similar results as we did when we pulled out the samples with an observed phone call for $\boldsymbol{\phi}$. Most of the estimates remain nearly the same as our starting values, but we observe more of a spread in the estimates. With respect to tolerance, we are more interested in how people act when they are friends ($X_t = 1$) than when they are not ($X_t = 0$). We see estimates for $\mathbf{A}_{(1,0)}$ move towards 0, and estimates for $\mathbf{A}_{(1,1)}$ move towards one. This is more evidence for tolerance being observed. When people are already friends, they become more likely to stay friends and less likely to break friendships. Although this only holds for the pairs where at least one call was observed, $\frac{121}{400}$ of these pairs only had one observed phone call over the 280-day period, which means there could have been 279 days where a pair remained friends without a phone call - supporting the idea of tolerance.

Recall that the definition of tolerance also included the notion of an agent being N-tolerant. This stated that "An agent that chooses to maintain an unrewarding connection for n turns is $n - tolerant$". We can also measure the estimated n-tolerance of each pair using the geometric distribution. We

are interested in finding the number of days that a pair of friends will remain friends before breaking the friendship.

Here, we consider each time-step between each pair of agents that are currently friends as a Bernoulli trial. We can make the independence assumption since we are only utilizing the resulting value of the algorithm. Thus instead of looking at the state of the Markov chain that have already been observed, we are using the estimates of $\mathbf{A}_{(1,0)}$ to predict behavior in the future, hence we have Bernoulli trials. We define the possible outcomes are either maintain the friendship or break the friendship, with $P(\text{Friendship Broken} = \mathbf{A}_{(1,0)})$. In the context of the geometric distribution, a 'success' is the friendship being broken, and $p = \mathbf{A}_{(1,0)}$. X is defined as the number of trials before a success occurs. To estimate n-tolerance, we will compute $E(X) = \frac{1}{p} = \frac{1}{\mathbf{A}_{(1,0)}}$ for each pair of individuals that had at least one call made. The results for the 400 samples are shown below:

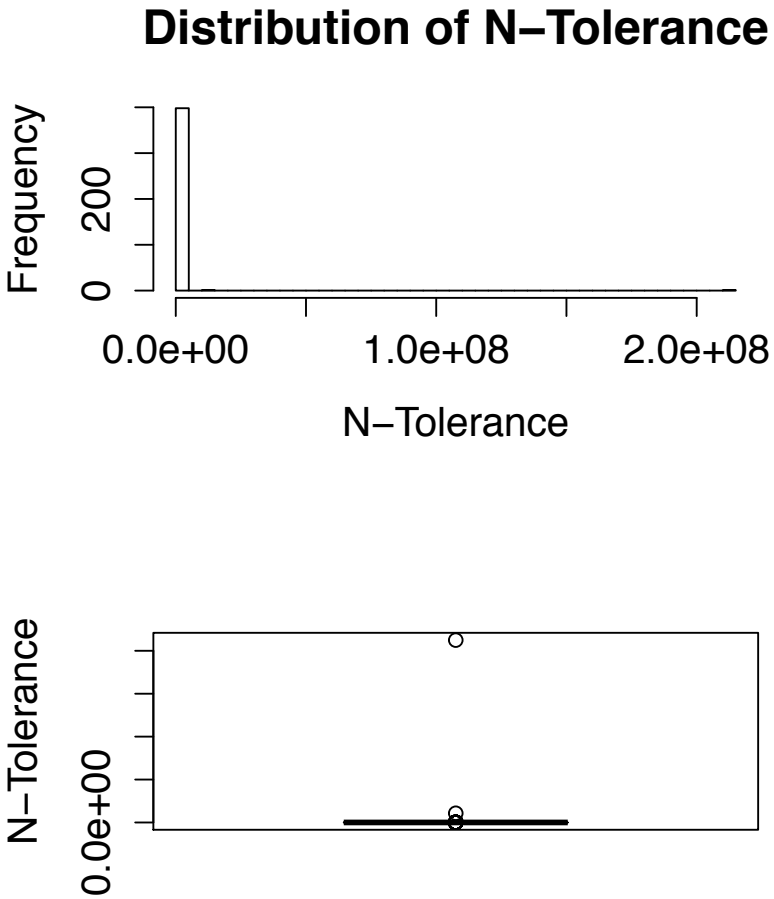

Figure 7.7: Distribution of N-Tolerance represented as a histogram and boxplot

Since $E(X) = \frac{1}{p}$, and $p \in [0.1]$, we have $E(X) \in (0, \infty)$. Therefore the outliers make the rest of the data much harder to observe. If we remove the farthest outlier (one pair of nodes), with an n-tolerance of 557,890 days, we obtain the following plots, which contain data from 399/400 pairs of nodes.

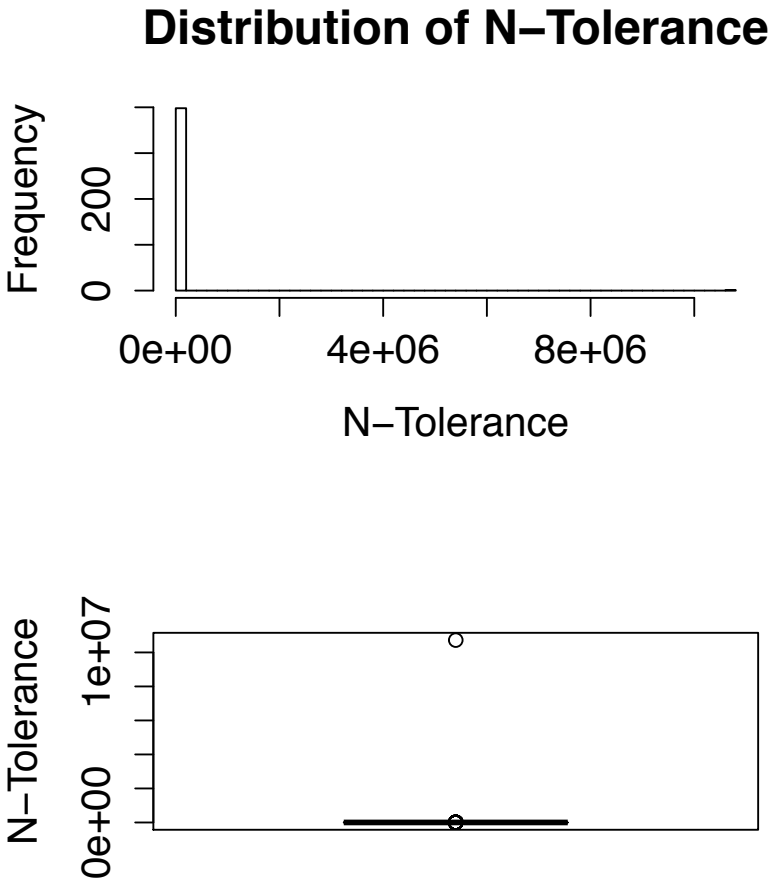

Figure 7.8: Distribution of N-Tolerance (1 outlier removed)

Again, we observe one more significant outlier (one pair of nodes), with an n-tolerance of 26,884 days, that we remove to observe the rest of the data, which contains 398/400 pairs of nodes.

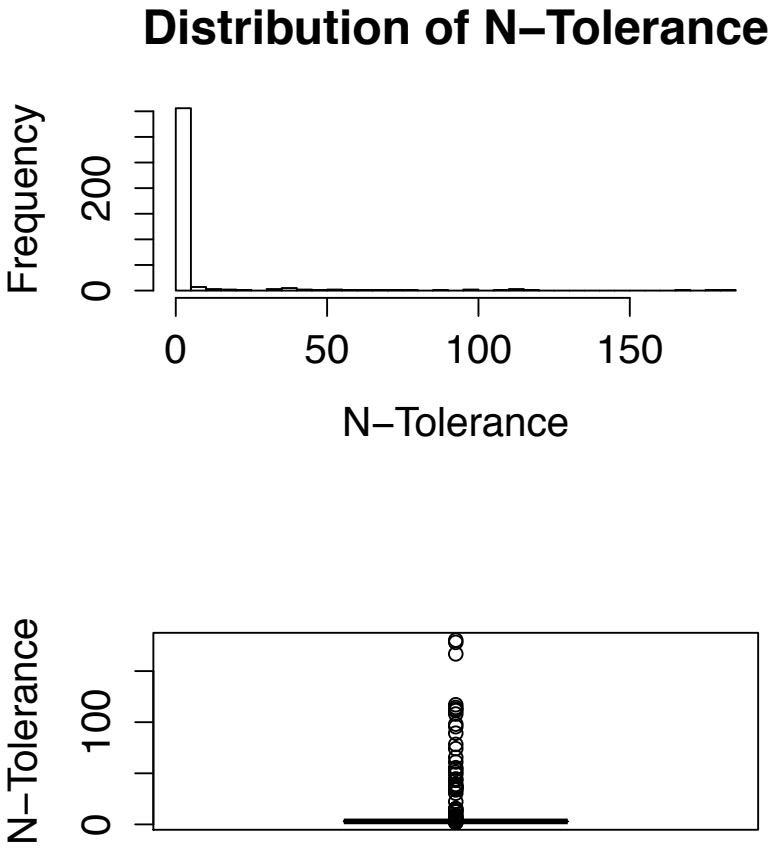

Figure 7.9: Distribution of N-Tolerance (2 outliers removed)

Now we can see that $\frac{398}{400}$ samples have estimated N-tolerance between 0 and 150 days, with the majority of these samples having an N-tolerance of less than 10. We now have evidence of the existence of N-tolerance, with the exception of two outliers, observed in the range $(0, 150)$ days. Note that by removing two outliers, we have removed two pairs of nodes. However, we have left 398 pairs of nodes intact so we have not eliminated a significant amount of data.

After examining the parameter estimates for the hidden Markov model, we conclude that there was some sort of change in the samples from their initial estimates to the final estimate observed. We also conclude that N-tolerance is a measurable attribute and was observed in our system. A natural

next step to take is determining if the change in the estimates was significant. This will tell us if, when there was enough evidence to make a change to the estimates, the change was big enough to matter.

7.3 Significance Tests

Since we have a large sample size (n=400) we can use a t-procedure to test for significance.

A paired t-test as well as a Wilcoxon signed-rank test was performed between the starting values for each of the HMM's and the ending values. Thus the first sample had the following form, which are the initial estimates for each of the parameters at the beginning of procedure:

	ϕ_0	ϕ_1	$A_{(0,0)}$	$A_{(0,1)}$	$A_{(1,0)}$	$A_{(1,1)}$
1	0.00176	0.00176	0.942	0.058	0.339	0.661
2	0.00176	0.00176	0.942	0.058	0.339	0.661
⋮	⋮	⋮	⋮	⋮	⋮	⋮
399	0.00176	0.00176	0.942	0.058	0.339	0.661
400	0.00176	0.00176	0.942	0.058	0.339	0.661

Table 7.1: Starting Values (First set of samples)

While the second sample consisted of the ending estimates for each of the 400 pairs with an observed phone call. Thus the second sample has the following form:

	ϕ_0	ϕ_1	$A_{(0,0)}$	$A_{(0,1)}$	$A_{(1,0)}$	$A_{(1,1)}$
1	0.01006	0.01009	0.942	0.058	0.3389	0.6610
2	8.37×10^{-10}	0.76387	0.98931	0.01069	0.02037	0.97962
⋮	⋮	⋮	⋮	⋮	⋮	⋮
399	6.707×10^{-3}	0.00673	0.942	0.058	0.33899	0.66100
400	6.708×10^{-3}	0.00672	0.942	0.058	0.33899	0.66100

Table 7.2: Ending Values (Second set of samples)

Therefore, 6 paired t-tests were performed, treating each column in the matrices as a sample for the test. The following results were obtained.

	t	df	P-value	$\overline{x}_{\text{difference}}$
ϕ_0	15.2	399	$< 2.20 \times 10^{-16}$	-0.01004
ϕ_1	7.92	399	2.41×10^{-14}	0.0776
$A_{(0,0)}$	4.46	399	1.06×10^{-5}	-0.00391
$A_{(0,1)}$	4.46	399	1.06×10^{-5}	0.00391
$A_{(1,0)}$	6.62	399	1.12×10^{-10}	-0.0315
$A_{(1,1)}$	6.62	399	1.12×10^{-10}	0.0315

Table 7.3: Paired T-test results

For this test, the null hypothesis is that there is no difference in the means of the two samples. In Table 8.4, t is the studentized t ratio obtained, df is the degrees of freedom of the test, the P-value is the probability of obtaining a test statistic at least as extreme as the one observed, and $\overline{x}_{\text{difference}}$ is the estimated change in mean from the starting sample to the ending sample.

These tests confirmed that the change in estimates for the samples with call observations was significant for all estimated parameters. However, we see that $\boldsymbol{\phi}$ as well as $A_{(1,0)}$ and $A_{(1,1)}$ are the most significant according to the p-value, which is what we hypothesized during our discussion of tolerance. Recall that $A_{(1,0)}$ and $A_{(1,1)}$ govern the state of friendship at $t+1$ when the pair are friends at time t. Since we have defined tolerance as maintaining an unrewarding connection, these are the two parameters that control the tolerance for a given pair.

This confirms what was expected. If a phone call is observed between two nodes, it is intuitive to think that their emission probabilities would be significantly different than the starting values, which are heavily weighted towards the pairs without any calls observed between them.

The Wilcoxon signed-rank test yielded the following results, which confirm the t-test results above.

	V	P-value
ϕ_0	3,376	$< 2.20 \times 10^{-16}$
ϕ_1	122	$< 2.20 \times 10^{-16}$
$A_{(0,0)}$	62,949	$< 2.20 \times 10^{-16}$
$A_{(0,1)}$	17,251	$< 2.20 \times 10^{-16}$
$A_{(1,0)}$	78,082	$< 2.20 \times 10^{-16}$
$A_{(1,1)}$	2,118	$< 2.20 \times 10^{-16}$

Table 7.4: Wilcoxon Sign-Rank test

Finally, we turn to testing the significance of our N-tolerance results. For this t-test, we used the null hypothesis that the N-tolerance observed is equal to what the N-tolerance would be for our initial estimates, $E(X) = \frac{1}{0.339} = 2.949853$. To mirror the process we used for visualizing the data, we performed three different t-tests on all of the results, the results with one outlier removed, and the results with two outliers removed. The following results were obtained.

	t	df	p-value	Observed Mean
All Data	1.0496	399	0.2947	557,890.6
1 Outlier Removed	1.0002	398	0.3178	26,884.58
2 Outliers Removed	7.591	397	2.28E-13	8.594

Table 7.5: N-Tolerance t-test results

Thus we see that in the first two cases, we do not obtain statistically significant results. However, we also see that the outliers were so large that they brought the sample mean of the group up to a number that really doesn't make sense. To put it in perspective, an n-tolerance of 26,884 corresponds to over 73 years; while 557,890 corresponds to over 1,528 years. In the context of friendships, these n-tolerances do not make sense.

Once we remove the outliers, however, we obtain very significant results as well as an n-tolerance that we can make sense of. We then conclude that the n-tolerance of this group is significantly different than the estimated n-tolerance that we started with.

7.4 Preliminary Implementation of HTR

We are now able to use these estimates in the implementation of the Highest Tolerated Reward rule. By integrating the estimated probabilities of certain actions, we are able to better simulate a real-world network such as the Social Evolution data studied.

We ran into issues when trying to integrate the two different systems due to their fundamental differences. The Hidden Markov Model uses real data, which consisted of two totally separate networks (friendship and phone calls). The Highest Tolerated Reward MAS is a simulation for the real data, but is implemented as one network and using phone calls as a reward.

Due to time constraints, we were unable to obtain full results for the HTR simulation. Currently, the output for the HTR simulation consists only of network attributes such as the clustering coefficient, and a table representing both connections and phone calls. Due to the differences between the representation of the real data and the representation of the simulation, we were unable to determine a plan of action for analysis of the simulation results against the real data within the time allotted for the project, and leave it as future work. We believe that with the proper formatting of either or both of the datasets, complete analysis will yield successful results.

However, we were still able to obtain positive preliminary results. By only implementing the estimate for $\boldsymbol{\phi}_1$, we were able to obtain simulated results with clustering coefficients very near to the observed clustering coefficient for the Social Evolution data. The clustering coefficient for the *SocializeTwicePerWeek* relationship over the entire time period was 0.6148, and over a preliminary simulation study of 30 trials, we obtained a range of clustering coefficients from .57 to .62, with an mean of 0.5863. For perspective, before adding the estimate of $\boldsymbol{\phi}$, the clustering coefficient for the

simulations was consistently less than 0.2. This leads us to believe that our results are promising in further adapting the HTR rule.

8 Conclusions

We are positive about the preliminary results from our initial Multi Agent System simulations. Achieving observed clustering coefficients so similar to the real data with only one adjustment leads us to believe there is a promising future in making our implementation of the Multi Agent System as realistic as possible. We continue to adjust inputs and procedure to achieve more desirable results.

Through the Hidden Markov Model results, we have successfully shown a viable way of observing tolerance in a dynamic social network. Through the combined results of the estimates of $A_{(1,0)}, A_{(1,1)}$ and $\boldsymbol{\phi}$, we conclude that tolerance has been observed in a real dataset. This is not only valuable for this paper, but for previous work in MultiAgent systems which have observed tolerance in simulations. This work has shown that tolerance is not just a side-effect of the decision rule chosen for the system, nor is it a purely sociological theory, but an observable trait of dynamic social networks in real-world situations.

9 Future Work and Limitations

9.1 Limitations of the Data

The nature of the data that we have used has caused some limitations within the project. The survey data used had only 5 observations over a nine-month period. Daily survey data would have been better suited for the model. The study was also done from 2008-2009, but was not made available until 2012. Since the study was performed, cell phone technology has made incredible advances and is more prolific now than it was when the data was published. The same study, if performed today, could likely yield more results with the utilization of social media applications and text messaging in addition to phone calls.

9.2 Statistical Assumptions

In our implementation of the HMM, we defined both $\mathbf{X}$ and $\mathbf{Y}$ as binomial variables. Making this assumption has caused us to lose some of the data, and thus is a limitation of the project. For instance, defining $\mathbf{Y}$ as a discrete variable representing the number of calls in a given day, rather than as a Bernoulli trial and defining $\mathbf{X}$ as a categorical variable representing the 'type' of friendship reported allows more of the data to be represented in the model.

Additionally, we have assumed independence between all samples. Since each sample represents a possible connection between any two agents, there is bound to be some sort of correlation between the samples. For instance, two possible connections that share a third are likely to have some sort of dependence.

A further limitation of the project is the Markov assumption. We have made the assumption that friendship at time t is only dependent on time $t-1$, when in reality, friendship is a much longer-term attribute. By finding a way to address this limitation, our results could be strengthened.

9.3 Continuation of Computational Model

A natural next step of this project is finishing the implementation of the HTR Neighborhood Evaluation Rule. Although we have seen positive preliminary results, we have yet to fully integrate the estimated parameters from the HMM into the system. We have also explored the idea of a theoretical value for optimal tolerance, but have yet to formalize the definition and proof.

Additionally, we have viewed tolerance in the simplest of ways - either tolerant or non-tolerant. In the real world, we would expect different levels of tolerance, which has also been introduced in a preliminary paper (Genicot 2011). Allowing agents to fall somewhere on a spectrum of tolerance would further enhance the real-world applicability of the model.

10 Appendices

10.1 Table of Variables

Variable Name	Meaning	Value	Page Reference
$p_{ij}(t)$	Payoff Function	0-1	14
$r_{ij}(t)$	Reward Function	0-1	14
w	Time-Discount Factor	0.5-1.0	14
$R_{ij}(t)$	Reward Average Function	0-1	14
θ	Seeking Threshold	0.5-0.9	16
α	Maintaining Threshold	0.1-0.9	
γ	Random Connection Probability	0-1	17
δ	Friend of a Friend Probability	0-1	17
$\mathbf{X}$	Friendship	Binomial	21
$\mathbf{Y}$	Phone Call	Binomial	21
$\mathbf{A}$	Transition Probability Matrix	0-1	21
$\boldsymbol{\pi}$	Initial State Probability Vector	0-1	21
$\boldsymbol{\phi}$	Emission Probability Vector	0-1	21
	Number of Nodes	84	24
	Number of Calls	3700	27
	Number of Directed Connections	400	27
	Number of Time-Steps	298	27

10.2 Data Formatting

The following steps were taken to clean the data:

1. Count number of calls, duration for each caller/receiver pair
2. Order by userID then by destID
3. Compute Average Duration (Total Duration/Frequency)
4. Compute Frequency Received (Number of times Receiver called Caller)
5. Create Factors for survey relationships
6. Combine call and survey data into one data frame
7. Remove entries where destID and userID match
8. Remove any entries with a negative value
9. Remove outlier entries determined by the following:
    - Frequency ≥ 75 per month with less than 3 days with a recorded phone call
    - TotDuration$\geq$5000 per month with less than 3 days with a recorded phone call

```
setwd("Thesis")
getwd()

library(plyr)
#First, we will create a table with all of the call data
#in a given survey period. We will then clean the survey data
#and finally combine the two.

#Read calls data (split by survey date) & Survey data
calls=read.csv("Thesis/calls.csv", header=TRUE)
survey=read.csv("Thesis/Surveydata.csv", header=TRUE)
#Order by user ID and then by destination user ID
calls<-calls[order(calls$user_id, calls$dest_user_id),]
#Create data frame that will be filled in containing condensed
#Information for all pairs
connections = data.frame(UserID=numeric(0), DestID=character(),
    NumOfDays=character()
                         , Frequency=character(), TotDuration=
                           character())
userID=calls$user_id[1]
destID=calls$dest_user_id[1]
```

```
numOfDays=298
callsCount=0
totDuration=0

#For each call
for(i in 1:length(calls$user_id)){
    #Check if there's another call with same user/dest pair
    #and increment calls count and duration
  if (isTRUE(all.equal(c(userID, destID),c(calls$user_id[i],
      calls$dest_user_id[i])))){
    callsCount=callsCount+1
    totDuration = totDuration+calls$duration[i]
  }
  #Otherwise, append new entry to connections frame
  else{
    if(callsCount !=0){
      connections=rbind(connections, c(userID, destID, numOfDays
          ,callsCount, totDuration))
      totDuration=calls$duration[i]
    }
    #Move to next call in file & repeat
    userID=calls$user_id[i]
    destID=calls$dest_user_id[i]
    callsCount=1
  }
}
#append final entry to connections
connections=rbind(connections, c(userID, destID, numOfDays,
    callsCount, totDuration))
#order by userID and then by destID
connections<-connections[order(connections[,1], connections[,2])
    ,]
#remove first entry (because it's nonsense)
connections<-connections[2:length(connections[,1])-1,]
#Compute average duration by total duration/frequency
#round to one decimal place
AvgDuration<-connections[,5]/connections[,4]
AvgDuration<-round(AvgDuration,1)
connections<-cbind(connections, AvgDuration)
#compute the frequency received for each grouping
#i.e. number of times destID called userID
frequencyRec<-c(rep(0,length(connections[,1])))
for(i in 1:length(connections[,1])){
  caller<- connections[i,1]
  receiver<- connections[i,2]
  receiverCalled<-which(connections[,1]==receiver)
  recIndex = which(connections[receiverCalled,2]==caller)
  if(length(recIndex)!=0){
    frequencyRec[i]=connections[receiverCalled[recIndex],4]
```

```
  }
}
connections<-cbind(connections, frequencyRec)
#tolerance - number of times userID called destID minus
#number of times destID called userID
tolerance<-connections[,4] - connections[,7]
#compute tolerance 'level' - either 1,-1, or 0
toleranceLevel<-tolerance
for(i in 1:length(toleranceLevel)){
  if(toleranceLevel[i]==0){
    toleranceLevel[i]=0
  }
  else if(toleranceLevel[i]<0){
    toleranceLevel[i]=-1
  }
  else {
    toleranceLevel[i]=1
  }
}
connections<-cbind(connections,tolerance,toleranceLevel)
#define column names for data frame
colnames(connections)<-c("UserID", "DestID", "NumOfDays", "
    Frequency", "TotDuration","AvgDuration","FrequencyRec","
    tolerance","toleranceLevel")
#We are now done with cleaning and formatting the calls data


#Now we will begin cleaning and formatting the survey data
#Order by idA then by idB
survey<-survey[order(survey$id.A, survey$id.B),]
#create factors for the different options for relationship
frel<-factor(survey$relationship)
frel<-factor(frel,levels(frel)[c(1,3,4,5,2)])
#collapse survey data for each pair into one entry with binary
#columns if they answered for that relationship or not
survey1<- ddply(survey, c('id.A', 'id.B'), function(x) c(count=
    nrow(x), blog=levels(frel)[1] %in% x$relationship, facebook=
    levels(frel)[2] %in% x$relationship,politics=levels(frel)[3]
    %in% x$relationship, socialize=levels(frel)[4] %in% x$
    relationship,closefriend=levels(frel)[5] %in% x$relationship)
    )
#We are now done with cleaning and formatting the survey data
# and can move on to combining the survey and calls

#Merges calls and survey into one table by CALLS
# won't include survey data from people who did not call
# each other
table<-merge(connections, survey1, by.x=c('UserID', 'DestID'),
    by.y=c('id.A', 'id.B'), all.x=TRUE)
```

```
#replaces NA's with zeros
table[is.na(table)]<-0
#removes any survey data in which the destID and userID match
self<-table$UserID==table$DestID
table<-table[self==FALSE,]
#removes calls with negative entries
negCalls<-table$TotDuration<0
table<-table[negCalls==FALSE,]

#Done.

# optional: write table to file
# write.csv(table, "Survey5Table.csv", row.names=FALSE, na="")
#Logistic regression
blog<-glm(formula=table$blog~table$Frequency + table$TotDuration
    + table$FrequencyRec, family="binomial", data=table)
facebook<-glm(formula=table$facebook~table$Frequency + table$
    TotDuration + table$FrequencyRec, family="binomial", data=
    table)
politics<-glm(formula=table$politics~table$Frequency + table$
    TotDuration + table$FrequencyRec, family="binomial", data=
    table)
socialize<-glm(formula=table$socialize~table$Frequency + table$
    TotDuration + table$FrequencyRec, family="binomial", data=
    table)
closefriend<-glm(formula=table$closefriend~table$Frequency +
    table$TotDuration + table$FrequencyRec, family="binomial",
    data=table)
summary(blog)
summary(facebook)
summary(politics)
summary(socialize)
summary(closefriend)
#CHI SQUARE P VALUES
#1-pchisq(614.62,466)
#[1] 4.272671e-06
#> 1-pchisq(616.74,466)
#[1] 3.259215e-06
#> 1-pchisq(627.94,466)
#[1] 7.470636e-07
#> 1-pchisq(580.19,466)
#[1] 0.0002381774
#> 1-pchisq(647.29,466)
#[1] 4.974709e-08
#> 1-pchisq(462.92,463)
#[1] 0.4923083
#> 1-pchisq(462.16,463)
#[1] 0.5022768
#> 1-pchisq(479.97,463)
```

```
#[1] 0.2833759
#> 1-pchisq(416.23,463)
#[1] 0.9417797
#> 1-pchisq(526.33,463)
#[1] 0.02195997
#nullDev<-c(614.62, 616.74, 627.94, 580.19, 647.29)
#> resDev<-c(462.82, 462.16, 479.97, 416.23, 526.33)
#> g<-nullDev-resDev


#Creates a table merging the other way around -- includes
# survey data for which there are no calls as well as
# survey data which includes calls
surveyTab<-merge(survey1, connections, by.x=c('id.A', 'id.B'),
    by.y=c('UserID', 'DestID'), all.x=TRUE)
surveyTab<-surveyTab[surveyTab$socialize==1,]
#Removes entries where id.A = id.B
surveyTab<-surveyTab[surveyTab$id.A!=surveyTab$id.B,]
```

10.3 Compute days without a call

```
calls=read.csv("calls.csv", header=TRUE)
calls<-calls[order(calls$user_id, calls$dest_user_id),]
library(chron)
dates<-calls[,2]
dates[1:10]
difftime(dates[1], dates[3000],units='days')
as.POSIXct(dates)
mydate = strptime(dates[1],format='%d/%b/%Y:%H:%M')
mydate
mydate<-as.character(dates[1])
mydate1 = strptime(mydate,format='%m/%d/%Y %H:%M')
mydate1
as.POSIXct(mydate1)


smallDates<-dates[50:100]
myDates<-strptime(smallDates, format='%m/%d/%Y %H:%M')
as.POSIXct(myDates)
max(myDates)
min(myDates)
difftime(max(myDates), min(myDates), units='days')

calls=read.csv("calls.csv", header=TRUE)
calls<-calls[order(calls$user_id, calls$dest_user_id),]
calls<-calls[calls$user_id!=calls$dest_user_id, ]
```

10.4 Create Visual Data

```
survey2<-read.csv("Survey2Table.csv", header=TRUE)
survey3<-read.csv("Survey3Table.csv", header=TRUE)
survey4<-read.csv("Survey4Table.csv", header=TRUE)
survey5<-read.csv("Survey5Table.csv", header=TRUE)
survey6<-read.csv("Survey6Table.csv", header=TRUE)
totSurvey<-rbind(survey2, survey3, survey4, survey5, survey6)

totSurvey<-totSurvey[totSurvey$TotDuration<5000,]
totSurvey<-totSurvey[totSurvey$Frequency<75,]
totSurvey<-totSurvey[totSurvey$AvgDuration<150,]
totSurvey<-totSurvey[totSurvey$FrequencyRec<75,]

survey2<-survey2[survey2$TotDuration<5000,]
survey2<-survey2[survey2$Frequency<75,]
survey2<-survey2[survey2$AvgDuration<150,]
survey2<-survey2[survey2$FrequencyRec<75,]
plotDist(survey2)

survey3<-survey3[survey3$TotDuration<5000,]
survey3<-survey3[survey3$Frequency<75,]
survey3<-survey3[survey3$AvgDuration<150,]
survey3<-survey3[survey3$FrequencyRec<75,]
plotDist(survey3)

survey6<-survey6[survey6$TotDuration<5000,]
survey6<-survey6[survey6$Frequency<75,]
survey6<-survey6[survey6$AvgDuration<150,]
survey6<-survey6[survey6$FrequencyRec<75,]
plotDist(survey6)


model<-glm(formula = closefriend ~ TotDuration + Frequency +
    FrequencyRec, family = "binomial", data = totSurvey)

plotDist<- function(totSurvey){
#FREQUENCY: Friends vs. Non Histogram
a<-totSurvey$Frequency[totSurvey$count==0]
b<-totSurvey$Frequency[totSurvey$count!=0]
par(mfrow=c(3,2))
hist(a, xlim=c(0,60), ylim=c(0,400), xlab="", col="blue", main="
    Frequency - Non-Friends")
hist(b, col=rgb(0, 1, 0, 0.5), xlab="", main="Frequency - 
    Friends")

#TOTAL DURATION: Friends vs. Non Histogram
a<-totSurvey$TotDuration[totSurvey$count==0]
b<-totSurvey$TotDuration[totSurvey$count!=0]
```

```
hist(a, xlim=c(0,4000), ylim=c(0,500), xlab="", col="blue", main
    ="Tot Duration - Non-Friends")
hist(b, col=rgb(0, 1, 0, 0.5), xlab="", main="Tot Duration - 
    Friends")

#FREQUENCY RECEIVED: Friends  vs. Non Histogram
a<-totSurvey$FrequencyRec[totSurvey$count==0]
b<-totSurvey$FrequencyRec[totSurvey$count!=0]
hist(a, xlim=c(0,60), ylim=c(0,500), xlab="", col="blue", main="
    Freq. Rec - Non-Friends")
hist(b, col=rgb(0, 1, 0, 0.5), xlab="", main="Freq. Rec - 
    Friends")
return
}
```

10.5 Network Visualization

```
#Read in SURVEY DATA
surveyTab<-read.csv("Thesis/Survey1.csv")
surveyTab<-surveyTab[surveyTab$relationship=="
    SocializeTwicePerWeek",]
friendGraph<-graph.data.frame(surveyTab)
#Read in CALL DATA
callTab<-read.csv("Thesis/Survey1Table.csv")
callGraph<-graph.data.frame(callTab)
#Plot Directed Networks
plot(friendGraph)
plot(callGraph)
#Plot Undirected Networks
friendGraph_sym<-as.undirected(friendGraph, mode='collapse')
plot(friendGraph_sym)
friendGraph_sym_layout <- layout.fruchterman.reingold(
    friendGraph_sym)
plot(friendGraph_sym, layout=friendGraph_sym_layout)
callGraph_sym<-as.undirected(callGraph, mode='collapse')
plot(callGraph_sym)
callGraph_sym_layout <- layout.fruchterman.reingold(callGraph_
    sym)
plot(callGraph_sym, layout=callGraph_sym_layout)
#degree distribution
deg_Friend_in <- degree(friendGraph, mode="in")
hist(deg_Friend_in)
deg_Friend_out <- degree(friendGraph, mode="out")
hist(deg_Friend_out)

deg_call_in <- degree(callGraph, mode="in")
hist(deg_call_in)
deg_call_out <- degree(callGraph, mode="out")
hist(deg_call_out)
```

10.6 Create Time Series

```
setwd("Thesis")
calls=read.csv("calls.csv", header=TRUE)
calls<-calls[order(calls$user_id, calls$dest_user_id),]
library(chron)
#Formates dates into a POSIX object that R recognizes as dates
calls[,2]<-as.POSIXct(strptime(calls[,2], format='%m/%d/%Y_%H:%M
    '))
#converts days to integer representations based on start of
    experiment
calls[,2]<-cut(calls[,2], breaks="day", labels=FALSE)
calls<-data.frame(calls)
#removes unkown user IDs and dest user IDs
calls<-calls[is.na(calls$dest_user_id_if_known)==FALSE,]
calls<-calls[is.na(calls$user_id)==FALSE,]
#removes duration and phone hash
calls<-calls[,c(1,2,4)]
#Removes entries where user id = dest id
self<-calls$user_id == calls$dest_user_id_if_known
calls<-calls[self==FALSE,]
calls<-calls[order(calls$user_id, calls$dest_user_id_if_known,
    calls$time_stamp),]
#create empty matrix to fill in time series
user<-sort(c(rep(seq(1:84),84)))
dest<-c(rep(seq(1:84),84))
edges<-cbind(user,dest)
lab<-colnames(edges)
empty<-c(rep(0,7056))
for(i in 1:298){
  edges<-cbind(edges, empty)
  lab<-c(lab, i)
}
colnames(edges)<-lab

for(i in 1:length(calls$user_id)){
  userID<-calls$user_id[i]
  destID<-calls$dest_user_id_if_known[i]
  date<-calls$time_stamp[i]
  edges[84*(userID-1)+destID,date]<-1
}

#Find initial probabilities
numCalls<-apply(edges[,3:300],1,sum)
sum(numCalls)/(7056*298)
write.csv(edges, "callsTS.csv", row.names=FALSE)
```

10.7 Compute Transition Probability Matrix

```
library(plyr)
survey<-read.csv("Thesis/Surveydata.csv")
#Restrict relationship to only close friend
survey<-survey[survey$relationship=="SocializeTwicePerWeek",]
#Factor the survey dates
fdate<-factor(survey$survey.date)
#Order survey data
survey<-survey[order(survey$id.A, survey$id.B, survey$survey.
    date),]
#Remove entries where id.A=id.B
self<-survey$id.A==survey$id.B
survey<-survey[self==FALSE,]
#create empty table of edges to be filled in
a<-sort(c(rep(seq(1:84),84)))
b<-c(rep(seq(1:84),84))
edgesSur<-cbind(a,b)
lab<-colnames(edgesSur)
empty<-c(rep(0,7056))
for(i in 1:6){
  edgesSur<-cbind(edgesSur, empty)
}
dim(edgesSur)
colnames(edgesSur)<-c("id.A","id.B",levels(fdate))
levels(fdate)
#Fill in friendships
for(i in 1:length(survey$id.A)){
  id.a<-survey$id.A[i]
  id.b<-survey$id.B[i]
  date<-which(levels(fdate)==survey$survey.date[i])
  edgesSur[84*(id.a-1)+id.b,date+2]<-1
}
dim(edgesSur)
#Compute transition probabilities
zerzer<-0
zerone<-0
onezer<-0
oneone<-0
#Compute top row of matrix
for(i in 3:7){
  plus<-edgesSur[edgesSur[,i]==0,i+1]
  zerzer<-zerzer+count(plus)[1,2]
  zerone<-zerone+count(plus)[2,2]
}
#compute second row of matrix
for(i in 3:7){
  plus<-edgesSur[edgesSur[,i]==1,i+1]
  onezer<-onezer+count(plus)[1,2]
```

```
  oneone<-oneone+count(plus)[2,2]
}
zer<-sum(zerzer,zerone)
one<-sum(onezer,oneone)
pi<-matrix(c(zerzer/zer,zerone/zer,onezer/one,oneone/one),nrow
    =2,byrow=TRUE)
pi

delta<-c(count(edgesSur[,3])[1,2]/length(edgesSur[,3]), count(
    edgesSur[,3])[2,2]/length(edgesSur[,3]))
delta
```

10.8 Run Expectation-Maximization Algorithm

```
library(HiddenMarkov)
# callsTS: Time-series for binomial calls (whether call happened
    on day or not) rows=pairs of users
# and columns are number of days in the study
callsTS<-read.csv("Thesis/callsTS.csv", header=TRUE)
callsTS<-data.matrix(callsTS)
# pi: Transition Probability Matrix for latent state (friendship
    )
# Dataset included six surveys spread over the study where
    participants responded
# about their friendship with others in the study. The estimates
    for pi were found by
# counting the observed probability a no stayed a no for the
    next survey (.942)
# a no turned into a yes (.058), a yes turned into a no (.339)
    and a yes stayed a yes (.661)
pi<-matrix(c(.942,.058,.339,.661), nrow=2, byrow=TRUE)
# delta: marginal probability distribution of hidden states at
    first time point
# estimates taken from original data (# of pairs that said yes/
    total # of pairs)
delta<-c(.876,.124)
# obs: observed binomial of time series representing whether or
    not there was a call
#     between two participants
obs<-callsTS[7000,3:300]
# pn: vector that represents n in the binomial distribution (n=
    number of days)
pn<-list(size=rep(1, length(obs)))
# x: discrete time hidden markov model object (dthmm)
# pm: list(prob=c(.00176, .00176)) (p for binomial distribution
    in each of the latent states)
#     Making assumption they are equal to begin with
#     Estimated by # of 1's in callsTS/total entries in callsTS
sum(callsTS[3,3:300])
#Go through and perform BaumWelch algorithm on every pair and
    record results
EMres<-matrix(ncol = 11, nrow = 0)
for (i in 1:length(callsTS[,1])){
  obs<-callsTS[i,3:300]
  x<-dthmm(obs,pi,delta, "binom", list(prob=c(.00176, .00176)),
      pn, discrete=TRUE)
  x.EM<-BaumWelch(x)
  EMres<-rbind(EMres, c(callsTS[i,1:2], x.EM$pm$prob, x.EM$Pi
      [1,], x.EM$Pi[2,], x.EM$iter, x.EM$LL,))
}
sums<-apply(callsTS[,3:300],1,sum)
```

```
EMres<-cbind(EMres,sums)
EMres<-EMres[,1:11]
write.csv(EMres, "Thesis/BaumWelchResults.csv", row.names=FALSE)
```

10.9 Significance Tests

```
estimates<-read.csv("Thesis/BaumWelchResults.csv")
#****EMISSION PROBABILITY GRAPHS******
par(mfrow=c(1,2))
hist(estimates[,3], main="All phi[0]", xlab="phi[0]", xlim=c
    (0,1))
hist(estimates[,4], main="All phi[1]", xlab="phi[1]", xlim=c
    (0,1))
#****EMISSION PROBABILITy GRAPHS NONZERO CALLS****
par(mfrow=c(1,2))
hist(estimates[estimates[,11]>0,3], xlim=c(0,1), ylim=c(0,350),
    main = "Nonzero calls phi_0",xlab="phi_0")
t.test(estimates[,3], estimates[estimates[,11]>0,3])
wilcox.test(estimates[,3], estimates[estimates[,11]>0,3])
hist(estimates[estimates[,11]>0,4], xlim=c(0,1), main="Nonzero
    calls phi[1]", xlab="phi[i]")
t.test(estimates[,4], estimates[estimates[,11]>0,4])
wilcox.test(estimates[,4], estimates[estimates[,11]>0,4])
par(mfrow=c(1,1))
hist(1-estimates[estimates[,11]>0,4], xlim=c(0,1), main="
    Probability of Tolerance", xlab="1-phi[1]")
boxplot(1-estimates[estimates[,11]>0,4],range=0, main = "
    Probability of Tolerance")
#****TRANSITION PROBABILITY GRAPHS******
par(mfrow=c(2,2))
hist(estimates[,5], main="(0,0) entry", xlab="Probabilities")
hist(estimates[,6], main="(0,1) entry", xlab="Probabilities")
hist(estimates[,7], main="(1,0) entry", xlab="Probabilities")
hist(estimates[,8], main="(1,1) entry", xlab="Probabilities")
#****TRANSITION PROBABILITY GRAPHS NONZERO PAIRS*****
sig<-estimates[estimates[,11]>0,3:8]
t.test(sig[,3],estimates[,5])
wilcox.test(sig[,3], estimates[,5])
hist(sig[,3], main="A[0,0] Nonzero calls", xlab="A[0,0]", breaks
    =10, xlim=c(0,1))
t.test(sig[,4],estimates[,6])
wilcox.test(sig[,4], estimates[,6])
hist(sig[,4], main="A[0,1] Nonzero calls", xlab="A[0,1]",breaks
    =10, xlim=c(0,1))
t.test(sig[,5],estimates[,7])
wilcox.test(sig[,5], estimates[,7])
hist(sig[,5], main="A[1,0] Nonzero calls", xlab="A[1,0]",breaks
    =10, xlim=c(0,1))
t.test(sig[,6],estimates[,8])
wilcox.test(sig[,6], estimates[,8])
hist(sig[,6], main="A[1,1] Nonzero calls", xlab="A[1,1]",breaks
    =10, xlim=c(0,1))
count(estimates[,11]==1)
```

```
start<-data.frame(matrix(rep(c(.00176, .00176,estimates[1,5:8]),
    length(sig[,1])),byrow=TRUE,ncol=6))
#**** TEST IF VALUES SIGNIFICANTLY CHANGED FROM STARTING VALUES***
t.test(as.numeric(start[,1]), as.numeric(sig[,1]), paired=TRUE)
wilcox.test(as.numeric(start[,1]), as.numeric(sig[,1]), paired=
    TRUE)
t.test(as.numeric(start[,2]), as.numeric(sig[,2]), paired=TRUE)
wilcox.test(as.numeric(start[,2]), as.numeric(sig[,2]), paired=
    TRUE)
t.test(as.numeric(start[,3]), as.numeric(sig[,3]), paired=TRUE)
wilcox.test(as.numeric(start[,3]), as.numeric(sig[,3]), paired=
    TRUE)
t.test(as.numeric(start[,4]), as.numeric(sig[,4]), paired=TRUE)
wilcox.test(as.numeric(start[,4]), as.numeric(sig[,4]), paired=
    TRUE)
t.test(as.numeric(start[,5]), as.numeric(sig[,5]), paired=TRUE)
wilcox.test(as.numeric(start[,5]), as.numeric(sig[,5]), paired=
    TRUE)
t.test(as.numeric(start[,6]), as.numeric(sig[,6]), paired=TRUE)
wilcox.test(as.numeric(start[,6]), as.numeric(sig[,6]), paired=
    TRUE)
#******** N -TOLERANCE USING GEOMETRIC DISTRIBUTION**********
    **
expected<-1/sig[,5]
par(mfrow=c(2,1))
hist(expected, breaks=50, main="Distribution␣of␣N-Tolerance",
    xlab="N-Tolerance")
summary(expected)
boxplot(expected , ylab="N-Tolerance")
t.test(expected, mu=(1/.339))
max(expected)
exp2<-expected[expected<100000000]
hist(exp2, breaks=50, main="Distribution␣of␣N-Tolerance", xlab="
    N-Tolerance")
summary(exp2)
boxplot(exp2 , ylab="N-Tolerance")
t.test(exp2, mu=(1/.339))
exp3<-expected[expected<10000000]
hist(exp3, breaks=50, main="Distribution␣of␣N-Tolerance", xlab="
    N-Tolerance")
summary(exp3)
boxplot(exp3 , ylab="N-Tolerance")
t.test(exp3,mu=(1/.339))
```

Bibliography

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] M. Boguna and R. Pastor-Satorras. Emergence of clustering, correlations, and communities in a social network model. *Physical Review*, 2004.

[3] C Cassisi, P Montalto, M Prestifilippo, M Aliotta, A Cannata, and D Patanè. Monitoring volcano activity through hidden markov model. In *AGU Fall Meeting Abstracts*, volume 1, page 2855, 2013.

[4] N. Christakis and J. Fowler. Social contagion theory: Examining dynamic social networks and human behavior. *Statistics in Medicine*, 2013.

[5] S. Currarini, M. Jackson, and P. Pin. An economic model of friendship: Homophily, minorities and segregation. *The Working Paper Series, Department of Economics, Ca'Foscari University of Venice*, 2007.

[6] J. Davidsen and H. Ebel. Emerge of a small world from local interaction: modeling acquaintance networks. *Physical Review Letters*, page 88, 2002.

[7] Jordi Delgado. Emergence of social conventions in complex networks. *Artificial Intelligence*, pages 171–175, 2002.

[8] W. Dong, K. Heller, and A. Pentland. Modeling the co-evolution of behaviors and social relationships using mobile phone data. *10th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, pages 134–143, 2011.

[9] W. Dong, K. Heller, and A. Pentland. Graph-coupled hmms for modeling the spread of infection. *Uncertainty in Artificial Intelligence*, pages 266–275, 2012.

[10] W. Dong, K. Heller, and A. Pentland. Modeling infection with multi-agent dynamics. *Social computing, Behavior-Cultural modeling, and Prediction (SBP)*, pages 172–179, 2012.

[11] N. Eagle, A. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *National Academy of Sciences of the United States of america*, pages 15274–15278, 2009.

[12] C. Fleming. What is communication? the definition of communication. *Communication*, 2011.

[13] B. Fosdick and P. Hoff. Testing and modeling dependencies between a network and nodal attributes. *EprintarXiv*, 2013.

[14] G. Genicot. Tolerance and compromise in social networks. *Georgetown University Preliminary Draft*, 2011.

[15] K Ivanova and Ivan Iordanov. Two-population dynamics in a growing network model. *Physica A: Statistical Mechanics and its Applications*, 2012.

[16] Matthew Jackson and Brian Rogers. Meeting strangers and friends of friends: How random are social networks? *The American Economic Review*, pages 890–915, 2007.

[17] K. Joseph, W. Wei, and K. Carley. An agent-based model for simulataneous phone and sms traffic over time. *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 66–68, 2013.

[18] Akash Krishnan and Matthew Fernandez. System and method for recognizing emotional state from a speech signal, October 24 2013. US Patent App. 14/062,433.

[19] P Krivitsky and Handcock M. A seperable model for dynamic networks. *Journal of the Royal Statistical Society*, 2014.

[20] J Kumpula and J Onnela. Emergence of communities in weighted networks. *Physical Review Letters*, 2007.

[21] P. Lapachelle. The use of scoial networking in community development. *CDPractice*, 2011.

[22] J Leezer and Y Zhang. Emergence of social norms in complex networks. In *Symposium on Social Computing Applications*, 2009.

[23] J. Levy and B. Pescosolido. *Social Networks and Health*. JAI Press, 2002.

[24] A Madan and D Lazer. Social sensing: Obesity, unhealthy eating, and exercise in face-to-face networks. *IN proceedings of Wireless Health*, 2010.

[25] A Madan and A Pentland. Sensing the 'health state' of a community. *Social Evolution Dataset*, 2012.

[26] M Marsili and F Vega-Rodondo. The rise and fall of a networked society: a formal model. *Proceedings of the National Academy of Sciences*, 2004.

[27] M. McPherson, L. Smith-Lovin, and J Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001.

[28] R.M. Milardo. Personal choice and social constraint in close relationships: Applications of network analysis. *Friendship and Social Interaction*, 1986.

[29] L. Milroy. *Social Networks, in the Handbook of Language Variation and Change*. Blackwell Publishing Ltd, 2008.

[30] T. Minsheng, M. Xinjun, Z. Guessoum, and Z. Huiping. Rumor diffusion in an interests-based dynamic social network. *The Scientific World Journal*, 2013.

[31] K Myunghwan and L Jure. Modeling social networks with node attributes using the multiplicative attribute graph model. *EprintarXiv*, 2011.

[32] Tiina Ojanen and Jelle Sijtsema. Intrinsic and extrinsic motivation in early adolescents' friendship development. *Journal of Adolescence*, 2010.

[33] J O'Malley and J Onella. Topics in social network analysis and network science. *Physics and Society*, 2014.

[34] Noa Pinter-Wollman, Elizabeth Hobson, and Jennifer Smith. The dynamics of animal social networks: analytical, conceptual, and theoretical advances. *EprintarXiv*, 2013.

[35] Toivonen Riita. A comparitive study of social network models: Network evolution models and nodal attribute models. *Social Networks*, 2009.

[36] Roche. Em algorithm and variants. *Service hospitalier Frederic Joliet*, 2012.

[37] V. Schwanda and N. Bazarovab. Relational maintenance on social network sites: How facebook communication predicts relational escalation. *Computers in Human Behavior*, 2014.

[38] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: Modeling, analysis and simulations. *AI*, 1997.

[39] T. Snijders. Introduction to dynamic social network analysis. *University of Gronigen*, 2005.

[40] T. Snijders, A. Lomi, and V. Torlo. A model for the multiplex dynamics of two-mode and one-mode networks, with an application to employment preference, friendship, and advice. *Physics and Society*, 2013.

[41] E. Stattner, M. Collard, and N. Vidot. D2snet: Dynamics of diffusion and dynamic human behaviour in social networks. *Computers in Human Behavior*, 2013.

[42] S. Valenzuela, N. Park, and K. Kee. Is there social capital in a social network site? facebook use and college students' life satisfaction, trust, and participation. *Journal of Computer-Mediated Communication*, 2009.

[43] Michael J. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley and Sons, 2009.

[44] Wayne Wu and Yu Zhang. Pattern analysis in dynamic social networks. *Mathematics and Computing, Bard College*, 2010.

[45] Shi Zhong and Joydeep Ghosh. Hmms and coupled hmms for multi-channel eeg classification. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, volume 2, pages 1254–1159, 2002.

[46] L. Zhoua, Li Dingb, and T. Fininc. How is the semantic web evolving? a dynamic social network perspective. *Computers in Human Behavior*, 2013.