Log Linear Models for Prediction and Analysis of Networks

A Dissertation
Submitted
to the Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

By
Vladimir Ouzienko
January, 2013

Examining Committee Members:

Dr. Zoran Obradovic, Advisory Chair, Department of Computer and Information Sciences

Dr. Vasilis Megalooikonomou, Department of Computer and Information Sciences

Dr. Alexander Yates, Department of Computer and Information Sciences

Dr. Avrum Gillespie, External Member, Department Of Medicine

ABSTRACT

The heightened research activity in the interdisciplinary field of network science can be attributed to the emergence of the social network computer applications. Researchers understood early on that data describing how entities interconnect is highly valuable and that it offers a deeper understanding about the entities themselves. This is why there were so many studies done about various kinds of networks in the last 10-15 years. The study of the networks from the perspective of computer science usually has two objectives. The first objective is to develop statistical mechanisms capable of accurately describing and modeling observed real-world networks. A good fit of such mechanism suggests the correctness of the model's assumptions and leads to better understanding of the network. A second goal is more practical, a well performing model can be used to predict what will happen to the network in the future. Also, such model can be leveraged to use the information gleaned from network to predict what will happen to the networks entities. One important leitmotif of network research and analysis is wide adaptation of log linear models. In this work we apply this philosophy for study and evaluation of log-linear statistical models in various types of networks.

We begin with proposal of the new Temporal Exponential Random Graph Model (tERGM) for the analysis and predictions in the binary temporal social networks. We then extended the model for applications in partially observed networks that change over time. Lastly, we generalize the tERGM model to predict the real-valued weighted links in the temporal non-social networks. The log-linear models are not limited to networks that change over time but can also be applied to networks that are static. One such static network is a social network composed of patients undergoing hemodialysis. Hemodialysis is prescribed to people suffering from the end stage renal disease; the treatment necessitates the attendance, on non-changing schedule, of the hemodialysis clinic for a prolonged time period and this is how the social ties are formed. The new log-linear Social Latent Vectors (SLV)

model was applied to study such static social networks. The results obtained from SLV experiments suggest that social relationships formed by patients bear influence on individual patients clinical outcome. The study demonstrates how social network analysis can be applied to better understand the network constituents.

ACKNOWLEDGMENTS

I heartily acknowledge the guidance and ongoing support of my advisor Dr. Zoran Obradovic. He introduced me to the world of scientific research for which I will be forever grateful. His help and encouragement were integral part of my continuous advancement in my research. Dr. Obradovic always made himself available to answer whatever questions I had and was ready to help to overcome whatever blocks I had stumbled upon. His vast academic experience and insight had helped me to progress from my first data mining graduate class to the point where my works were published.

My many special thanks to Dr. Yuhong Guo, who expanded my horizons in mathematics and machine learning, her help was essential in making my first break-through as scientist. I also would like to thank Dr. Slobodan Vucetic, immensely talented teacher and researcher, his teaching is what immersed me into the machine learning field. Thank you to Dr. Justin Shi, who made me feel welcome at the department.

I am very thankful to Dr. Vasilis Megalooikonomou and Dr. Alexander Yates for serving on my graduate committee and for providing useful advise and feedback. A very special thanks to Dr. Avrum Gillespie, my committee's external reviewer, our scientific collaboration was intriguing and exciting, it allowed me to learn so many things outside of my scientific domain. Thank you to Dr. Heather Hammer, my other interdisciplinary collaborator, for sharing her vast experience and knowledge with me.

Many thanks to my colleague and friend, Kosta Ristovski, for many hours we had spent together going over mathematical problems.

Finally, I would like to thank my friends at the department, unraveled group of smart people: Nemanja Djuric, Mihajlo Grbovic, Vuk Malbasa and Vladan Radosavljevic for making my years at the Computer and Information Sciences Department a pleasant experience.

To my wife Anna.

To my parents Stase and Anatoliy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The idea of using network analysis to study natural processes had gained a lot of traction in recent years. Such analysis especially became popular in social science [2], and were largely facilitated by spread of Internet based social applications. A social network describes the interactions (relationships) between participating social entities (actors). The well known web applications, Facebook and MySpace, are examples of social networks, where each social actor is a person and two persons can be linked together if there are interactions between them (e.g., email exchanges). Another example of social networks is the informal relationship graph of farming estates [41], where each social actor is a family that owns the farm. In this farming community network, a social visit constitutes a link between two social entities. Actors and links are two essential elements of social networks regardless of their semantics while directed graphs are the main representation and analytical tool of social network analysis (SNA). A social network can be either static or dynamic. To analyze temporal dynamic social networks, one needs to investigate the evolving patterns of the networks along the time axis, including the network trends, the changes of actor's roles, and the strengthening or weakening of the relationships. The typical prediction problems associated with temporal social networks are inferences of links or actor roles, determining the strengths of the relationships and imputation of the missing links.

While temporal SNA is a well studied subject [22, 26], the topic of prediction in such structures is less explored. Many published works in the field such as [81] investigate the social networks from the perspective of statistical inference. Their objective is to infer the most probable statistics which would explain the network evolution process. In Chapter we demonstrate how log linear models can be applied for the prediction in temporal networks.

A special case of temporal social networks is network surveys which have proven to be invaluable tools for social scientists. In such surveys often a group of people from an

enclosed social setting (e.g. classroom, village etc.) is asked to identify the people of the same group they think of as a friend. The social network observations which are done over time on the same set of people are called panel surveys and each survey conducted at any given time t is called a wave panel. In practice, not all respondents always choose to provide answers to such surveys, therefore the social scientists are forced to deal with missing data. The adverse effects of non-responsive actors in social network surveys were studied extensively in the past. The general consensus is that missing network information or complete absence of an actor from the network surveys will negatively affect the estimation of network properties [12], and underestimation of the network ties' strength [17]. Thus the heightened sensitivity of SNA to missing data as compared to less structured, non-network datasets raises the importance of accurate imputation models. To that end we introduce a new log linear approach for imputation in the social networks in Chapter .

In Chapter we demonstrate the versatility of log linear approach in the study of socially linked hemodialysis population in Philadelphia, PA. People who suffer from end-stage renal disease have to regularly visit an outpatient clinic for hemodialysis where they share personal thoughts and opinions about their disease and its treatment. Sharing thoughts about success or failure of a kidney transplant, or sharing dislikes of a strict dietary regiment may influence a patient's prognosis. Our objective was to identify probably social interactions of the hemodialysis patients and use that information to enhance mortality predictor accuracy by introducing a new log linear technique.

The network science is not limited to the study of the social networks, it is a much larger field. It is interdisciplinary science covering networks in engineering, climate science, biology and so on. One of the domains studied by networks science is medicine. The recent work [40] put forward the idea of using network analysis to study Phenotype Disease Network (PDN). PDN is a network where each node represents a human ailment and links between the nodes could be a prevalence rate of two co-morbid diseases or the statistical strength of their co-occurrence. Hidalgo at al. [40] proposed to construct PDN

from the hospitals' admissions data. The hospitals in the United States report patients' diagnosis using standardized codes for the insurance claim purposes. Therefore it is possible to tabulate the set of reported diagnosis within a group of hospitals for a set time period and count how many patients were diagnosed with ailments pairs. It was suggested that PDN based on hospitals admission data collected at regular time intervals exhibit temporal trends and changes [40]. One likely reason for change in statistical strength between diseases or change in their prevalence rate is that patients diagnosed with a set of diseases "travel" along the links of PDN. This is due to the fact that serious disease often progresses and new diagnoses are reported, hence creating causality link [40]. Also, certain diagnosis are seasonal in nature, and can be related to pollen or cold weather or any other environmental factors. The work by Hidalgo et al [40] also suggested that illness progression varies between patients groups stratified by race and gender, and highly connected diseases were found to be significantly correlated with higher mortality. The aspect we explore in Chapter is whether the temporal PDN can be statistically modeled akin to how temporal social networks were modeled int the past [83]. We posit that any statistical model capable of predicting the next temporal step of the PDN network more accurately than trivial baselines can provide a helpful insight into PDN evolution mechanism. The practical aspect of such model perhaps could be a part of the planning strategy for the health care providers.

CHAPTER 2

LOG LINEAR APPROACH FOR PREDICTION OF TEMPORAL SOCIAL NETWORKS

Most published work on predictions of temporal social networks is focused on link predictions. In particular, the time a specified group of social interactions will occur is predicted in a recent study [55]. Another recently studied temporal prediction problem is inferencing network structures at the next yet unseen step [66, 90]. The social network link imputation problem is also considered for data stored in relational databases [71]. These models predict how the network evolves. However, the attribute values of the participating actors are typically not considered (an exception is the method proposed in [71]). The prediction models employed in previous studies of temporal social networks are solely based on network structures and their topologies, albeit the temporal aspects of the social networks are always considered. However, in many cases, attribute values of actors are also available in social networks and are useful for accurate link prediction. Moreover, predicting the attribute values of participating actors could be as important as predicting the network graphs.

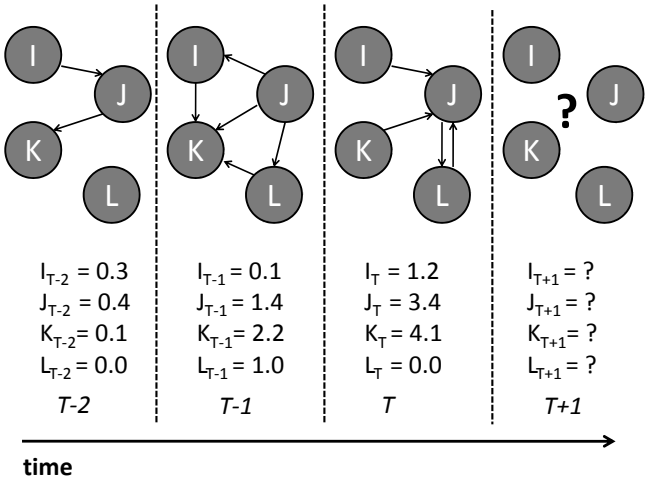

Figure 1: A conceptual representation of the prediction task

Given the evolving structure of a temporal network and the changing non-static attribute values of the network actors, the goal is to predict the network structure and actor values at the next unobserved time step. In Figure 1 we show a graphical representation of the prediction task. This figure demonstrates how the relationship graph of the invariant set of actors and actors' attributes are changing as time progresses from time steps $T-2$ to T. Then based on this historical data, we want to predict the relationship graph and attribute value of each actor at unobserved time step $T+1$. Instead of training a single joint probability prediction model for this prediction task, we build two conditional exponential random graph models, based on the observation over Gibbs sampling inference. These two conditional predictors are mutually dependent on each other, and can then be used to predict the network structures, i.e., the links, and the attribute values in an alternative way. Our empirical study suggests that this novel approach achieves better performance than baseline approaches.

A typical representation model for analysis of temporal social networks consists of a series of dichotomous adjacency matrices (also called sociomatrices) which define the states of a set of participating agents at each given observation time [31]. For example, to represent T temporal observations of k actors we will use T adjacency matrices, $N^1 \cdots N^T$, where each entry $N^t_{ij} = 1$ indicates the presence of a link between the actor i and actor j at time step t; conversely $N^t_{ij} = 0$ indicates the absence of such a link. For example, in a social network over the friendships of a class of students, a sociomatrix N^t is used to indicate the links between the students based on their friendship status at time t. The entry $N^t_{ij} = 1$ indicates that student i considered student j as his/her friend at time t. It is important to note that a relationship matrix does not have to be symmetric. In this example we can have $N^t_{ij} = 1$ and $N^t_{ji} = 0$ at the same time, which means that while student i considered student j as his/her friend, student j did not reciprocate the feelings. Moreover, the actors should not have self-referenced relations, thus diagonals of matrices $N^1 \cdots N^T$ should be always populated with zeros.

In the past, the p^* class of log linear statistical models [93], also known as Exponential Random Graph Models (ERGMs) [18, 31, 82] have been successfully applied to analyze and describe the sociomatrices discussed above. The ERGM is a log-linear model and is expressed as:

$$P(N) = \frac{1}{Z(\boldsymbol{\theta})} exp\{\boldsymbol{\theta}^{'} \mathbf{u}(N)\} \tag{1}$$

which defines the probability of a given social network N from the set $\mathcal{N}$ of all possible social networks. Here $\boldsymbol{\theta}^{'}$ is a transposed parameter vector, $\boldsymbol{u}(N)$ is a vector of sufficient statistics of the network N, and $Z(\boldsymbol{\theta})$is the normalization constant. An extended version of ERGM, the Temporal Exponential Random Graphical Model (tERGM), is proposed in [37], which specifically deals with the temporal aspect of social network analysis. This temporal model takes the Markovian assumption that each network matrix N^t , i.e. the observation matrix at the time t , is conditionally independent of all other prior observations $N^1 \cdots N^{t-2}$ given its immediate prior observed matrix N^{t-1}, which is:

$$P(N^t | N^{t-1}, N^{t-2} \cdots N^1) = P(N^t | N^{t-1}) \tag{2}$$

Thus the joint distribution in tERGM can be expressed as:

$$P(N^{1:T} | \boldsymbol{\theta}) = P(N^1) \prod_{t=2}^{T} P(N^t | N^{t-1}, \boldsymbol{\theta}) \tag{3}$$

where the conditional transition distribution is an extension of the log-linear model in (1), and can be expressed as

$$P(N^t | N^{t-1}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta}, N^{t-1})} exp\{\boldsymbol{\theta}^{'} \boldsymbol{\psi}(N^t, N^{t-1})\} \tag{4}$$

Here, $\boldsymbol{\psi}$ is a function of $\mathbb{R}_{k \times k} \times \mathbb{R}_{k \times k} \rightarrow \mathbb{R}^l$, where k is number of actors, which defines the statistics, i.e., the features. In [37], four statistics are defined, including *density*, *stability*,

reciprocity, and *transitivity*:

$$\psi_D(N^t, N^{t-1}) = \frac{1}{k-1}\sum_{ij}^{k} N_{ij}^t \tag{5}$$

$$\psi_S(N^t, N^{t-1}) = \frac{1}{k-1}\sum_{ij}^{k}[N_{ij}^t N_{ij}^{t-1} + (1-N_{ij}^t)(1-N_{ij}^{t-1})] \tag{6}$$

$$\psi_R(N^t, N^{t-1}) = k\frac{[\sum_{ij}^{k} N_{ji}^t N_{ij}^{t-1}]}{[\sum_{ij}^{k} N_{ij}^{t-1}]} \tag{7}$$

$$\psi_T(N^t, N^{t-1}) = k\frac{\sum_{pqr}^{k} N_{pr}^t N_{pq}^{t-1} N_{qr}^{t-1}}{\sum_{pqr}^{k} N_{pq}^{t-1} N_{qr}^{t-1}} \tag{8}$$

where k is number of actors (all statistics are scaled to be in $[0; k]$ range). The conditional probability function in (4) is a function of the parameters, $\boldsymbol{\theta}= \{\theta_D, \theta_S, \theta_R, \theta_T\}$, which correspond to the statistics $\{\psi_D, \psi_S, \psi_R, \psi_T\}$. Thus θ_D controls the density of the network, i.e., the number of existing links. θ_S controls the stability, or whether a link (or its absence) at time step $t-1$ continues to exist at time step t. θ_R drives the reciprocity, which is the degree that a link presented at time $t-1$ from actor i to j will result in a reciprocal link from j to i at time t. θ_T governs the transitivity, which is the propensity of the links from p to q and from q to r at $t-1$ resulting in a transitive link from r to p at time t.

Parameters of the tERGM are estimated from the sequence of temporal network observations $N^1 \cdots N^T$ using the Markov Chain Monte Carlo (MCMC) techniques [37, 82]. After the model parameters were learned, it is possible to make prediction over the network structure at future time step $T+1$ by applying Gibbs sampling method [82], which samples networks according to the conditional model given in (4) and the previous observation N^T.

The tERGM model introduced above considers only the structures and topologies of the temporal networks, while actor attributes are ignored. However, when the actor attributes are available and useful to know, one needs to make predictions over both the network structures and the attribute values in future steps. For instance, in the student friendship

network example, the Grade Point Average (GPA) of each student can be easily obtained at each observation step. Thus, in addition to an array of sociomatrices, one can have T numbers of $k \times 1$ attribute vectors $\boldsymbol{x}^{1:T}$ where $\boldsymbol{x}^i$ contains the GPA for all the students at time step i. The GPAs could be a factor that affects the friendships among students. Thus, it is reasonable to assume that the network structures and the attribute values are interactively dependent. Our proposed work is based on such a dependence assumption over the links and the attributes. Our method extends tERGM by building two mutually dependent conditional models which are used to jointly predict both N^{T+1} and $\boldsymbol{x}^{T+1}$.

2.1 Related Work

Temporal networks research is mostly applied to the domains of genetics and social network analysis. In genetics, a predominant question addressed by the research community is construction of the genes pathways (networks) based on temporal observation of gene expressions. A number of interesting models have been developed in this field [2, 51]. The question answered by these models is fundamentally different from our research subject. Genetic temporal models recover the network structure, based on underlying time series data. Our temporal model predicts future network structure and actor attributes based on time series of networks and actor attributes. Despite such a fundamental difference, network recovery models consider both links and data, and therefore we pay close attention to these models.

Contrary to genetics, the main question of the temporal SNA community is link prediction. There are two camps in this field. An overwhelming majority of publications on this topic predicts the network structure solely based on network topology [55, 66, 90]. The other camp also predicts links, but considers the actor attributes. Reported results that pursue such an approach were mostly concerned with specific types of datasets such as academic collaboration or business co-operation networks [9]. These datasets are textually rich, and therefore specialized models were applied to them. To the best of our knowledge, the SNA community has not yet considered simultaneously predicting data and links for

temporal networks.

The brief review of the previous work in this chapter covers two groups of models. The first group of models are "related models" that cannot be readily applied to the problem we are trying to solve, but at the same time are important to us because we drew upon them when we develop our approach. The second type, "baseline models", are link prediction models which can be applied in a temporal setting. Most of these baseline models do not predict actor attributes. We use them as baselines in our link predictions experiments on temporal social networks.

2.1.1 Related Models

A novel Hidden Temporal Exponential Random Graph Model (htERGM) was introduced [35] to recover latent temporal network structures based on attributes of observed nodes (actors). Specifically, it is shown how to recover the temporal network of the *Drosophila* gene expressions from the observations over the gene expression levels (node attributes). This approach incorporates the learning of both network structures and node (gene) attributes in a single, combined htERGM, expressed as

$$P(N^{1:T}, \boldsymbol{x}^{1:T}|, N^0) = \prod_{t=1}^{T} P(N^t|N^{t-1}, \boldsymbol{\theta}) P(\boldsymbol{x}^t|N^t, \Lambda) \tag{9}$$

Note that equation (9) consists of two conditional models. The first part is the transition model given in (4), which defines how the gene network evolves over time. The second log linear model, shown in (10)

$$P(\boldsymbol{x}^t|N^t, \Lambda) = \frac{1}{Z(\eta, N^t)} exp\left\{\eta \sum_{ij} \Phi(x_i^t, x_j^t, N_{ij}^t, \Lambda_{ij})\right\} \tag{10}$$

is called an emission model, and it defines the dependency of the node attributes over the underlying network topology. The Λ in Equation (10) is a time invariant activation function specific to Drosophila dataset. This function provides the degree of mutual activation or

suppression or activation between two genes and has $[-1, 1]$ range. Presence of the activation function Λ is a rather strong assumption, which might be perfectly valid for the gene expression network, but perhaps is inappropriate for the temporal social networks. Since two models, transition and emission, are involved such that one needs to learn two sets of unknown parameters: $\boldsymbol{\theta}$ and η , while the activation function Λ is directly estimated from the training dataset. Parameters $\boldsymbol{\theta}$ and η are treated as latent variables in htERGM and an Expectation-Maximization (EM) method is used for training. This approach is computationally slow so applications were limited to the retrieval of subnetworks of up to 10 genes. The new approach that we propose in this article learns two sets of model parameters separately which makes the learning process much faster and applicable for larger networks.

A follow up work to [35] was presented in [2], which introduced an algorithm named TESLA to address the slowness of the htERGM. In this new approach, the latent model is simplified to a convex temporal smoothed l_1 regularized logistic regression problem

$$P(x^t|\boldsymbol{\theta}^t) = exp\left(\sum_{i\in V}\theta^t_{ii}x^t_i + \sum_{(i,j)\in E^t}\theta^t_{ij}x^t_i x^t_j - A(\boldsymbol{\theta}^t)\right) \tag{11}$$

where $A(\boldsymbol{\theta}^t)$ is the log partition function, V is the set of nodes and E^t is a set of edges at time t. TESLA first learns parameters $\boldsymbol{\theta}$ for each node for all available time steps $t = 1 \cdots T$. Given $\boldsymbol{\theta}$, a simple sign function is applied to determine whether a link is present between nodes i and j at time t. However, in TESLA the transition model is omitted. It can only be applied to recover unknown structures of the temporal network based on the observed node values $\boldsymbol{x}^t$ for $t = 1 \cdots T$. Direct predictions of the future values for $\boldsymbol{x}^{t+1}$ and the edge set E^{t+1} are impossible in this model.

Another approach [81] models the network dynamics as a stochastic actor-driven process where each change occurs at one micro-step at a time. This approach models the changes of actors links as a function of node covariates (or 'actors covariates') and charac-

teristics of pairs of nodes ('dyadic covariates'). It takes the Markovian assumption of the network evolution process and posits that changes in the links are actor driven. Its main characteristic is modeling of the network events (appearance or disappearance of a link) as occurring at infinitesimal time intervals where only one actor gets the opportunity to change one link at a time. The modeling process works as follows, the actor is chosen for a given micro time-step who gets the opportunity to change one of her outgoing links. The choice of actor can be made either with equal probability among all actors or with probability corresponding to the actor's standing in the network. After the actor is chosen she gets the opportunity to change one outgoing link with probability proportional to the objective function. The objective function consists of the network, actor and dyadic covariates effects. To simulate the network evolution the process is repeated until the convergence of the estimation parameters. The stochastic actor-based model is used by researchers to study which of the network effects, or covariates, are prominent factors in evolution of the network. In our work we are more concerned with the predictive power of our proposed approach and other methods. The actor-driven stochastic model has some similarities with our methodology such as the Markovian assumption of the network evolution and utilizing log linear functions to model probabilities. The main difference between our model and [81], is that [81] looks at a longitudinal process and models latent link changes in continuous time as taking place between the actual observations, whereas our model only considers the discrete time steps when the network was observed.

The previous work [85] compares discrete vs. continuous time approach. It asserts that discrete observations do not capture the full dynamic of the network evolution because there is a possibility that there could have been many unobserved link changes in-between the network surveys. In [85] an example is cited where the network surveys done at times t and $t + 1$ recorded the creation of a link between two actors with similar characteristic (both non-drinkers in this example). Appearance of such a link in any discrete model is unequivocally classified as a homophily selection. The possibility here is that we did not

observe an interim process between times t and $t+1$ where one actor had become a drinker, which caused the other actor to start a therapeutic relationship with him (thus creating the outgoing link). This relationship had influenced the actor-drinker to stop consuming alcohol, so at the time $t+1$ we finally observe two non-drinkers in a social relationship. The fact that homophilic selection had nothing to do with what had actually transpired went completely unnoticed. The time-continuous approach is a plausible modeling solution for longitudinal networks, but missing element here is that it cannot be verified with real life datasets available today. The only way to validate the model assertion that there was an unobserved activity between the surveys would be to retroactively ask network participants a follow up questions. Unfortunately such retrospective studies are very rare and prone to error [10]. Nevertheless, the question of whether a discrete model can deal with unobserved changes has to be addressed. One way to address it is to determine whether the studied temporal network has too many link changes between observations which would suggest that it is unsuitable for treatment by a discrete model. A social network which evolves too fast between the survey times would not be a good candidate because of the increased likelihood that self-canceling link creations/deletions were not recorded. In [81] it was proposed to use the Jaccard index to measure the network change rate between the surveys. This measure calculates the rate of change between two observations by

$$\frac{C_{11}}{C_{11} + C_{01} + C_{10}} \tag{12}$$

where C_{11} is count of links present in both network, C_{01} is number of newly created ties and C_{10} is number of ties which were terminated. A low value of this index, less than 0.2, suggests that network is undergoing rapid change and it is likely that surveys missed a lot of activity. An index value close to 1.0 tells us that not much has happened between two observations. In our experiments we used two synthetic and two real life datasets, which are described in great detail in the Chapter . We calculated the Jaccard index for all our

datasets to ensure that networks are not changing too fast and are suitable for our model. The rate change for synthetic *Dataset1* and *Dataset2* averaged 0.39 and 0.30 respectively with a small variation of the index along the time axis. For the real life datasets *Delinquency* and *Teenagers* the average values were 0.40 and 0.35, also with very little variations. These values suggests that changes in temporal networks used in our experiments are adequately gradual and therefore our datasets are good candidates for treatment by a discrete model such as ours.

2.1.2 Baseline Models

A comprehensive description of a static graph link prediction algorithm which can also be applied to temporal network link prediction problem was recently published [46]. The article [46] introduced a new approach which enhances the accuracy of temporal network link prediction by combining a time-series approach with time-invariant algorithms. We provide brief descriptions of these methods because we will use them as baselines in our experiments on real life and synthetic datasets. The output of these algorithms is always score matrix S, where each entry $S(i, j)$ assigns the link occurrence score proportional to the predicted probability of the link from actor i to actor j at time step $T + 1$. Contrary to our approach, these methods do not exploit information available as attributes of actors.

Finally, here we also describe an alternative approach based on tensor factorization which predicts both links and attributes.

In order to predict links at $T + 1$, the time invariant link prediction algorithms reduce series of sociomatrices $N^1 \cdots N^T$ to a time invariant adjacency matrix $M^{1:T}$, where $M^{1:T}(i, j) = \sum_{t=1}^{T} N^t(i, j)$. The matrix $M^{1:T}$ is further reduced to the binary matrix M^* where $M^*(i, j) = 1$ if $M^{1:T}(i, j) > 1$, and 0 otherwise. The time invariant link prediction algorithms usually use such matrix representation.

The *Common Neighbor* algorithm assigns the probabilistic score for each entry in the score matrix S as the number of common neighbors shared by each possible pair of actors.

In a matrix form: $S = M^* * M^*$. *Common Neighbor* exploits the notion of transitivity; the higher count of common neighbors shared by two actors, the more likely these actors will be connected. The concept of transitivity as applied to social relationships was thoroughly studied in [39].

The *Adamic-Adar* method [1] is a measure similar to *Common Neighbor*. Adapted for the link prediction problem, *Adamic-Adar* assigns the link score between actors i and j as

$$S(i,j) = \sum_{k \in V, (i,k) \in E(M^*), (j,k) \in E(M^*)} \frac{1}{log(d(k))} \tag{13}$$

where d is degree of the actor. The *Adamic-Adar* measure is well-known in information retrieval domain.

The *Katz* measure [50] is based on a principle similar to *Common Neighbor* and *Adamic-Adar*. However, it also considers paths beyond length 2 between pairs of actors. The *Katz* sums paths of all possible lengths between two actors and exponentially dampens the longer paths to give them less weight than to the shorter paths. For the purpose of links prediction

$$S(i,j) = \sum_{l=1}^{\infty} \beta^l ||paths_{i,j}^{<l>}|| \tag{14}$$

where l is the length of the paths and β is a dampening parameter which has to be estimated from the training data. In [66] it was shown that a score matrix based on *Katz* measure can be derived as:

$$S = (I - \beta M^*)^{-1} - I \tag{15}$$

The *Preferential Attachment* method sets the link score in matrix S as the product of the degrees of participating actors , namely $S(i,j) = d(i) * d(j)$, where function d is the actor's degree. The *Preferential Attachment* algorithm assumes that the highly connected actors are more likely to be linked. This assumption is based on a preferential attachment phenomenon discovered in real-world networks [8].

An application of the *autoregressive integrated moving average* ARIMA model [14] is also proposed for the link prediction in temporal social networks [46]. Each possible link (i, j) between actors i and j during observations $t = 1 \cdots T$ can be viewed as a time series of the link occurrence frequency. The ARIMA model then can be fitted by exploring its parameter space $p = 0, 1, 2, 3; d = 0, 1; q = 0, 1, 2, 3$ where p is the number of autoregressive terms, d is the number of nonseasonal differences and q is the number of lagged forecast errors in the prediction equation. To determine the quality of the model the *Akaike information criterion* AIC measure is used [4, 46]. The model with the lowest AIC score is selected to predict the link frequency at time $T + 1$. This prediction is defined as $\hat{N_{ij}}^{T+1}$, and the prediction error as $sd(\hat{N_{ij}}^{T+1})$. The link occurrence score of matrix S is populated with probabilities of the link frequency at $T + 1$ to be greater than 1: $S(i, j) = Pr(\hat{N_{ij}}^{T+1} > 1)$. The time series model [46] is simple in the sense that it does not consider the temporal networks' topology. It only looks into a link's occurrence frequency independently from other actors. Therefore, its concern is the temporal nature of the link occurrence, whereas models described in the previously described baselines were only considering the network's topology and did not consider the temporal aspect.

The same study [46] exploits orthogonality of time invariant link prediction models and a time-series model by combining score matrices produced by both to yield a more accurate predictor. Authors introduced the *Hybrid Time Series Link Prediction Algorithm* which combines the score matrix generated by any of the time invariant algorithms (*Common Neighbor, Preferential Attachment , Adamic-Adar* and *Katz*) and time-series algorithm ARIMA. Namely, the new score matrix is derived as

$$S(i, j) = (S_S(i, j) + \frac{ms}{\alpha}) * (S_T(i, j) + \frac{mt}{\alpha}) \tag{16}$$

where α is a parameter greater than 1, S_S is normalized score matrix outputted by any of the static graph link prediction algorithms, S_T is a normalized score matrix generated by time-series link prediction algorithm and ms and mt are the minimum nonzero scores from

corresponding matrices. The comprehensive set of experiments on two real-life datasets demonstrated the advantage of this approach. Experiments have shown that a combination of time-series *ARIMA* and time invariant *Katz* yield the most accurate combined model.

A robust and innovative approach by [25] based on CANDECOMP/PARAFAC (CP) tensor decomposition [19] avoids the loss of the temporal information. Its output is also a score matrix S, but instead of collapsing the temporal networks into the time time-invariant adjacency matrix it takes the three-way tensor representation $\mathbb{Z}$ (the size of $k \times k \times T$, where k is number of actors) of the temporal network. We define $\mathbb{Z}(i, j, t) = 1$ if there was a link from actor i to actor j at time t and $\mathbb{Z}(i, j, t) = 0$ otherwise. Given such a tensor its L components CP decomposition is given by

$$\mathbb{Z} \approx \sum_{l=1}^{L} \lambda_l \boldsymbol{a}_l \circ \boldsymbol{b}_l \circ \boldsymbol{c}_l \tag{17}$$

The symbol $\circ$ denotes the outer vector product, λ_l is a positive real number, $\boldsymbol{a}_l$ and $\boldsymbol{b}_l$ are k-size real value vectors and $\boldsymbol{c}_l$ is real value vector of size T. The CP tensor decomposition is analogous to the Singular Value Decomposition (SVD) with some notable differences (for more details see [25]). This approach is using the extracted components to assign a score proportional to the likelihood of future link appearance to each pair of actors. The outer product of vectors $\boldsymbol{a}_l$ and $\boldsymbol{b}_l$ captures the relationship between the actors in the component l, the temporal interactions are stored in vectors $\boldsymbol{c}_l$. Authors propose a simple heuristic to account for the temporal activity stored in temporal profiles $\boldsymbol{c}_l$ by averaging the scores for the last 3 observations. Therefore the score matrix S_L for L-component decomposition is calculated as

$$S_L = \sum_{l=1}^{L} \gamma_l \lambda_l \boldsymbol{a}_l \circ \boldsymbol{b}_l \tag{18}$$

where

$$\gamma_l = \frac{1}{T_0} \sum_{t=T-T_0+1}^{T} \boldsymbol{c}_l(t) \tag{19}$$

and T_0 is customarily set to 3. Here, it is impossible to determine the number of components L "a priori". Instead, authors use an ensemble approach where the score matrices S_L are calculated for various values of $L \in \{l, l_{+1}, l_{+2} \dots l_{max}\}$ and the final matrix S is composed as

$$S = \sum_{L \in \{l, l_{+1}, l_{+2} \dots l_{max}\}} \frac{S_L}{||S_L||_F} \tag{20}$$

where $||S_L||_F$ is the Frobenius norm of the score matrix S_L. Besides the link prediction, the tensor decomposition can also be used for the prediction of actors' attributes. In our experiments we use the tensor decomposition (implementation provided by MATLAB Tensor Toolbox [7]), as a baseline for both link and attribute predictions.

2.2 The Proposed Model

A new efficient approach to make mutual predictions over the social network structure and actor attributes based on the historical data is described in this chapter. Specifically, given the social network link observations $N^1 \cdots N^T$, where N^t is a binary $k \times k$ sociomatrix, and the actor attribute observations $\boldsymbol{x}^1 \cdots \boldsymbol{x}^T$, where $\boldsymbol{x}^t$ is a k-length vector and k is a number of participating actors, we aim to make accurate predictions of the N^{T+1} and $\boldsymbol{x}^{T+1}$ in the future step.

Note that the network structure and the attribute values in a social network are mutually dependent on each other. Learning a joint distribution over them will end up predicting each of them alternatively using derived conditional models from the joint model. Based on this observation, we propose to learn directly two interdependent conditional prediction models, link prediction model, and actor prediction model, which can then be used to predict the network structure and attribute values interdependently to avoid the expensive

inference in htERGM. Our overall model will be called extended tERGM (etERGM), since the conditional models are still formulated under the similar framework as in tERGM.

2.2.1 Actor Attribute Prediction Model

For actor attribute prediction we use the following log-linear model:

$$P(\boldsymbol{x}^t|\boldsymbol{x}^{t-1}, N^t, \boldsymbol{\gamma}) = \frac{1}{Z(\boldsymbol{x}^{t-1}, N^t, \boldsymbol{\gamma})} exp\{\boldsymbol{\gamma}^{'}\boldsymbol{\psi}(\boldsymbol{x}^t, \boldsymbol{x}^{t-1}, N^t)\}\mathbb{N}(\boldsymbol{x}^t) \tag{21}$$

It describes the transition of attributes from time $t-1$ to time t, conditioning on the network structure N^t at time t. Here, $Z(\boldsymbol{x}^{t-1}, N^t,\boldsymbol{\gamma})$ is a normalization constant, and $\mathbb{N}(\boldsymbol{x}^t)$ is the regularization prior. As the basis for the prior, we used Gaussian multivariate distribution. The mean and covariance for our Gaussian regularized prior are estimated from training data. We choose Gaussian as a prior because of its smoothening effect on actor attributes along the time axis and thus inhibiting the oscillation of the predicted values. Our choice of prior regularizes the attribute predictions such that they stay within the range of their domain.

This model encodes the dependency of the attribute values $\boldsymbol{x}^t$ over the network structure represented by N^t in a direct way. The transposed model parameter $\boldsymbol{\gamma}^{'}$ is a vector corresponding to the statistic vectors $\boldsymbol{\psi}(\boldsymbol{x}^t,\boldsymbol{x}^{t-1}, N^t)$ which encodes the dependencies between the links among actors and theirs attributes. We used three statistics ψ_{links}, ψ_{sim}, and ψ_{dyads}:

$$\psi_{links}(N^t, \boldsymbol{x}^t) = k\frac{\sum_{i<j}^{k} N_{ij}^t N_{ji}^t \mathbb{I}(|x_i^t - x_j^t| < \sigma)}{\sum_{i<j}^{k} N_{ij}^t N_{ji}^t} \tag{22}$$

$$\psi_{sim}(\boldsymbol{x}^t, \boldsymbol{x}^{t-1}) = \sum_{i}^{k} \mathbb{I}(x_i^t, x_i^{t-1}, \sigma) \tag{23}$$

$$\psi_{dyads}(N^t, \boldsymbol{x}^t) = k\frac{\sum_{ij}^{k} N_{ij}^t \mathbb{I}(|x_i^t - x_j^t| < \sigma)}{\sum_{ij}^{k} N_{ij}^t} \tag{24}$$

To derive the ψ_{links} statistics we exploited the homophily effect often found in social network, also stated as "birds of feather flock together". We tested our assumption on two real life datasets. For each dataset we measured the average of the absolute values of the actor attributes differences, defined as $|x_i^t - x_j^t|$, for three distinct cases. We measured the average distance between the actors that are not connected, i.e. $N_{ij}^t = N_{ji}^t = 0$; that are partially connected by a single link: $N_{ij}^t = 1$ or $N_{ji}^t = 1$; and that are fully connected: $N_{ij}^t = N_{ji}^t = 1$. We discovered that fully connected actors on average have lesser absolute attribute difference than actors who are connected partially or not connected at all. Based on this discovery, we defined the ψ_{links} statistics as the ratio of the count of the fully connected pairs where actors attributes are similar, to the count of the fully connected pairs in graph N^t. We deem actor's i and j attributes similar if $|x_i^t - x_j^t| < \sigma$ where σ is a parameter and is estimated from the training data. In our experiments the training data was normalized to one standard deviation and we customarily set $\sigma = 0.3$. For example, if the number of actors is $k = 50$, the total number of fully linked pairs is 40 and out of those 10 are between actors with similar attributes, the value of ψ_{links} is $50 \times \frac{10}{40} = 12.5$.

ψ_{sim} captures temporal stability of actors attributes. If actors attributes do not change between the observations, ψ_{sim} is large and is small otherwise. $\mathbb{I}$ is the indicator function which returns 1 if actor value at times t and $t-1$ is similar, i.e $|x_i^t - x_i^{t-1}| < \sigma$ and returns 0 otherwise. For example, if $x_i^t = 0.6$, $x_i^{t-1} = 0.4$ and parameter σ is set to 0.3, we increase ψ_{sim} by 1.

Statistics ψ_{dyads} measures the similarity of attributes for the connected actors. It is a fraction of the total count of the linked pairs, which have similar attributes (as defined by $\mathbb{I}(x_i^t, x_j^t, \sigma)$) to the total count of all linked pairs of the graph. For example, if the number of actors is $k = 50$, the total number of links in the network N^t is 25 and out of those there are 12 links between actors having similar attributes (as defined by indicator function $\mathbb{I}$), the value of ψ_{dyads} is $50 \times \frac{12}{25} = 24$.

All three statistics are scaled such that their values are always in the $[0; k]$ range, where

k is the number of actors.

2.2.2 Link Prediction Model

The links are predicted by a log-linear model defined as

$$P(N^t|N^{t-1}, \boldsymbol{x}^t, \boldsymbol{\theta}) = \frac{1}{Z(N^{t-1}, \boldsymbol{x}^t, \boldsymbol{\theta})} exp\{\boldsymbol{\theta}^{'} \boldsymbol{\psi}(N^t, N^{t-1}, \boldsymbol{x}^t)\} \tag{25}$$

Similar to the tERGM model, this link prediction model defines the transition from N^{t-1} to N^t . However, different from before, we incorporate the dependency of N^t over the attributes $\boldsymbol{x}^t$ into the model directly.

In this log-linear model, $\boldsymbol{\psi}(N^t, N^{t-1}, \boldsymbol{x}^t)$ denotes a list of statistics. Here, we reused the four statistics already used at tERGM, which were shown in Equations (5), (6), (7), (8). Statistics defined in Equations (5), (6), (7), (8) capture only dependencies between matrices N^{t-1} and N^t and they do not link attribute values to the network. Thus, we define the additional statistics to capture such linkage ψ_{links} - the same statistics that were used in the actor attribute prediction model.

2.2.3 Learning Algorithm

The actor attribute prediction model and link prediction model proposed in and are both log-linear models. Two sets of parameters, $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$, need to be learned there.

The parameter $\boldsymbol{\theta}$ of the link prediction model consists of five coefficients $\{ \theta_D, \theta_S, \theta_R, \theta_T, \theta_{links} \}$ corresponding to the statistics $\{\psi_D, \psi_S, \psi_R, \psi_T, \psi_{links}\}$ respectively. To learn $\boldsymbol{\theta}$ we can apply Newton's optimization method demonstrated in [37] in straightforward fashion. The algorithm works as following: the $\boldsymbol{\theta}$ parameter is randomly initialized to a sensible value, then in an iterative manner it approximates the expectation using Gibbs sampling [31], followed by an update of parameter $\boldsymbol{\theta}$ such that it increases log-likelihood function of the observed temporal network. The algorithm repeats approximation with

updated parameters until updates to $\boldsymbol{\theta}$ become very small (we reach convergence). The Gibbs sampling of sociomatrices used in learning link prediction model parameters was well described in [82] and here we provide its brief description. To draw sociomatrix samples from posterior distribution initial matrix $N^{(1)}$ is chosen, and each element of this matrix is stochastically updated. The updating algorithm circles through the matrix $N^{(1)}$ defining the Markov chain stochastic process which asymptotically approximates required random graph distribution. At each update step only one element of the matrix is considered and stochastically updated, therefore the matrices $N^{(u)}$ and $N^{(u+1)}$ differ in one element at update step u. If element $N_{ij}^{(u)}$ at time u is being updated, then the probability of this element being 0 or 1 is defined by this conditional distribution:

$$P_{\boldsymbol{\theta}}(N_{ij}^{(u+1)} = a)|N^{(u)}) = P_{\boldsymbol{\theta}}(N_{ij} = a|\ N_{hk}^{(u)}\ \ \forall (h,k) \neq (i,j)) \quad (a = 0, 1) \tag{26}$$

The update process works as following, for a given sociomatrix N define two sociomatrices N^{ij0} and N^{ij1}, where both matrices are exact copies of matrix N, except the i, j element of matrix N^{ij0} and N^{ij1}, defined respectively as N_{ij}^{ij0} and N_{ij}^{ij1}, are set to: $N_{ij}^{ij0} = 0$, $N_{ij}^{ij1} = 1$. The conditional distribution from Equation (26) is defined as

$$logit\{P_{\boldsymbol{\theta}}(N_{ij}^{(u+1)} = 1|\ N_{hk}^{(u)}\ \ \forall (h,k) \neq (i,j)\ \ )\} = \boldsymbol{\theta}'(\psi(N^{ij1}) - \psi(N^{ij0})) \tag{27}$$

The Equation (27) defines the Gibbs sampling process for the time-invariant sociomatrix. We redefine Equation (27) for our link prediction model as:

$$logit\{P_{\boldsymbol{\theta}}(N_{ij}^{t,(u+1)} = 1|\ N_{hk}^{t,(u)}\ \ \forall (h,k) \neq (i,j)\ \ , N^{t-1}, \boldsymbol{x}^t)\} = \\ \boldsymbol{\theta}'\{\boldsymbol{\psi}(N^{t,(u),ij1}, N^{t-1}, \boldsymbol{x}^t) - \boldsymbol{\psi}(N^{t,(u),ij0}, N^{t-1}, \boldsymbol{x}^t)\} \tag{28}$$

We set i, j element of sampled sociomatrix $N^{t,(u+1)}$ to 1 at update step u with probability defined in Equation (28), while leaving all other element of the matrix intact. Sampling

one link at a time requires recalculation of model statistics, which can be slow. To speed up the sociomatrix sampling process the techniques "big update" and "inversion step" were described in [82]. The "big update" technique specifies the update of large set of cells of the sampled sociomatrix instead of the single link. The "inversion step" technique randomly inverts the whole sampled sociomatrix with a very small probability. It has been shown that "big update" speeds up the sampling process and "inversion step" leads to better Markov chain mixing [82]. We implemented both techniques in our approach.

Learning $\boldsymbol{\gamma}$ coefficients $\{\gamma_{links}, \gamma_{sim}, \gamma_{dyads}\}$ of actor attribute prediction model which correspond to the statistics $\{\psi_{links}, \psi_{sim}, \psi_{dyads}\}$ is done similarly to learning $\boldsymbol{\theta}$ of the link prediction model. Let,

$$L(\boldsymbol{\gamma}; \boldsymbol{x}^1, \cdots \boldsymbol{x}^T) = logP(\boldsymbol{x}^2, \boldsymbol{x}^3, \cdots, \boldsymbol{x}^T | \boldsymbol{x}^1, N^1 \cdots N^T, \boldsymbol{\gamma}) \tag{29}$$

$$M(t, \boldsymbol{\gamma}) = \mathbb{E}_{\boldsymbol{\gamma}}[\boldsymbol{\psi}(\underline{\boldsymbol{x}}^t, \boldsymbol{x}^{t-1}, N^t) | \boldsymbol{x}^{t-1}, N^t] \tag{30}$$

$$C(t, \boldsymbol{\gamma}) = \mathbb{E}_{\boldsymbol{\gamma}}[\boldsymbol{\psi}(\underline{\boldsymbol{x}}^t, \boldsymbol{x}^{t-1}, N^t) \boldsymbol{\psi}(\underline{\boldsymbol{x}}^t, \boldsymbol{x}^{t-1}, N^t)' | \boldsymbol{x}^{t-1}, N^t] \tag{31}$$

where expectations are taken based on samples from random variable $\underline{\boldsymbol{x}}^t$. We note, that

$$\bigtriangledown L(\boldsymbol{\gamma}; \boldsymbol{x}^1, \boldsymbol{x}^2, \cdots \boldsymbol{x}^T) = \sum_{t=2}^{T} (\boldsymbol{\psi}(\boldsymbol{x}^t, \boldsymbol{x}^{t-1}, N^t) - M(t, \boldsymbol{\gamma})) \tag{32}$$

and

$$\bigtriangledown^2 L(\boldsymbol{\gamma}; \boldsymbol{x}^1, \boldsymbol{x}^2, \cdots \boldsymbol{x}^T) = \sum_{t=2}^{T} (M(t, \boldsymbol{\gamma}) M(t, \boldsymbol{\gamma})' - C(t, \boldsymbol{\gamma})) \tag{33}$$

Similarly to [37] for learning link prediction model we apply Newton's optimization procedure to learn parameters $\boldsymbol{\gamma}$. The following procedure is used to approximate the expectations and update parameter values so that the log likelihood function defined by Equation (29) is increased:

1. Randomly initialize $\boldsymbol{\gamma}$

2. For $i = 1$ up until convergence

3. For $t = 2, 3, \cdots T$

4. Sample $\hat{\boldsymbol{x}}_{(i)}^{t,1}, \cdots \hat{\boldsymbol{x}}_{(i)}^{t,C} \sim P(\underline{\boldsymbol{x}}^t | \boldsymbol{x}^{t-1}, N^t, \boldsymbol{\gamma}^i)$

5. $\hat{\boldsymbol{\mu}}_{(i)}^t = \frac{1}{C} \sum_{c=1}^{C} \boldsymbol{\psi}(\hat{\boldsymbol{x}}_{(i)}^{t,c}, \boldsymbol{x}^{t-1}, N^t)$

6. $\hat{C}_{(i)}^t = \frac{1}{C} \sum_{c=1}^{C} \boldsymbol{\psi}(\hat{\boldsymbol{x}}_{(i)}^{t,c}, \boldsymbol{x}^{t-1}, N^t) \boldsymbol{\psi}(\hat{\boldsymbol{x}}_{(i)}^{t,c}, \boldsymbol{x}^{t-1}, N^t)'$

7. $\hat{H}_{(i)} = \sum_{t=2}^{T} [\hat{\boldsymbol{\mu}}_{(i)}^t \hat{\boldsymbol{\mu}}_{(i)}^{t\,\prime} - \hat{C}_{(i)}^t]$

8. $\boldsymbol{\gamma}^{(i+1)} \leftarrow \boldsymbol{\gamma}^{(i)} - \alpha \hat{H}_{(i)}^{-1} \sum_{t=2}^{T} [\boldsymbol{\psi}(\boldsymbol{x}^t, \boldsymbol{x}^{t-1}, N^t) - \hat{\boldsymbol{\mu}}_{(i)}^t]$

The parameter C in this algorithm specifies the number of samples drawn from posterior distribution. A greater value of C provides a finer update of parameter $\boldsymbol{\gamma}$ and usually achieves faster convergence. Also, for brevity, we omit the burn-in parameter. Burn-in specifies the number of samples that had to be thrown out for the good MCMC chain mixing. The main modification of our algorithm as compared to [37] is line 4. In a previous study [37], Gibbs sampling was used to sample from conditional distribution of sociomatrices. Here, we replaced Gibbs sampling with the Metropolis-Hastings algorithm to sample from $P(\underline{\boldsymbol{x}}^t | \boldsymbol{x}^{t-1}, N^t, \boldsymbol{\gamma}^i)$ distribution. Parameter α shown at line 8 specifies the learning rate of the optimization procedure.

For high dimensional data (large number of actors), it is useful to use the block-at-a-time technique discussed in [38]. In fact, this technique was implemented in our experiments.

2.2.4 Inference Algorithm

The goal of learning the etERGM is to make predictions over the social network structure and attribute values at time step $t + 1$; that is to infer the N^{t+1} and $\boldsymbol{x}^{t+1}$. Since the actor prediction model and the link prediction model are interdependent, we developed the following iterative algorithm to predict the network structure and actor attributes alternatively in a Gibbs sampling manner:

1. Randomly initialize $\hat{\boldsymbol{x}}^{t+1}$, $\hat{N}^{t+1}$

2. $iter = 1$

3. Do while $iter < maxiterations$

4. Randomly pick and sample one link (u, v): $\hat{N}^{t+1,(u,v)} \sim P(\underline{N}^{t+1}|N^t, \hat{\boldsymbol{x}}^{t+1}, \boldsymbol{\theta})$

5. Set $\hat{N}^{t+1}(u, v) = \hat{N}^{t+1,(u,v)}$

6. Randomly pick and sample one attribute (w): $\hat{\boldsymbol{x}}^{t+1,(w)} \sim P(\underline{\boldsymbol{x}}^{t+1}|\boldsymbol{x}^t, \hat{N}^{t+1}, \boldsymbol{\gamma})$

7. Set $\hat{\boldsymbol{x}}^{t+1}(w) = \hat{\boldsymbol{x}}^{t+1,(w)}$

8. if all attributes in $\hat{\boldsymbol{x}}^{t+1}$ were updated

9. add $\hat{\boldsymbol{x}}^{t+1}$ to $\mathbb{X}$

10. if all links in $\hat{N}^{t+1}$ were updated

11. add $\hat{N}^{t+1}$ to $\mathbb{N}$

12. $iter = iter + 1$

13. $\boldsymbol{x}^{t+1} = mean(\mathbb{X})$

14. $S = \sum_{\hat{N} \in \mathbb{N}} \hat{N}$

The algorithm starts by randomly initializing the attributes $\boldsymbol{x}^{t+1}$ and sociomatrix N^{t+1}. We initialize the sociomatrix N^{t+1} such that its link density is equal to the density of N^t. In lines 4-7 of the algorithm we iteratively sample one link and one attribute at a time in a Gibbs sampling manner. In line 4 of the algorithm we sample one randomly selected link and use the resulting updated sociomatrix as input into the sampling distribution on line 6. In line 6 we repeat the process by sampling the attribute value of one randomly selected actor and input the updated vector of attributes into the sampling distribution in line 4. At each iteration we check if all attributes or links were updated by our sampling process. If they were, we add sampled actors attributes vector to the collection $\mathbb{X}$ and sociomatrix to the collection $\mathbb{N}$. We continue collecting samples of sociomatrices and actors attributes until we reach the maximum number of iterations. Our prediction for the actors attributes

is mean of the collected samples (line 13). Our score matrix of the link probabilities is the sum of the collected sociomatrices (line14). Our algorithm is in essence a Gibbs sampling from a joint posterior distribution. We sample one link and one attribute at a time but to speed up the sampling process it is also possible to use a "block-at-a-time technique [38] for actors' attributes as well as a "big update" and "inversion step" techniques for sociomatrices [82].

2.2.5 Convergence

In the previous chapter we described the etERGM's inference algorithm. Its main idea is an iterative substitution of the drawn samples into interlocking probability distributions. The inference algorithm is intuitive by itself but we need to ensure that our inference procedure indeed converges. To evaluate convergence property of our algorithm we applied the standard convergence measurement for ERGM's [82]. It works as following: a) estimate the ERGM's parameter vector $\boldsymbol{\theta}$, b) draw the multiple samples from the estimated model, c) for each drawn sample calculate the model's statistics ψ_k, d) estimate t-ratio as

$$t_k = \frac{E_\theta(\psi_k) - \psi_0}{SD_\theta(\psi_k)} \tag{34}$$

where ψ_0 is the actual value of the statistic. It was suggested [82] that $|t_k| \leq 0.1$ indicates an excellent convergence, $0.1 < |t_k| \leq 0.2$ is good and $0.2 < |t_k| \leq 0.3$ is fair.

To calculate t-ratios for each statistic of both prediction models we trained etERGM on a single transition step from $t = 2$ to $t = 3$ of the *Delinquency*, a real life dataset which will be described in the experiments. Using estimated model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ we ran our inference algorithm. After the sufficient burn-in we sampled 1000 sociomatrices and 1000 vectors of the actors attributes. In Table 1 we report average differences between the simulated statistics and true values, the standard deviations of the simulated statistic and statistics' t-ratios.

Applying convergence criteria from [82], the results on average are fair or close to fair.

Table 1: Convergence estimates of the inference algorithm on time steps $t = 2, 3$ of the real life dataset *Delinquency*

Statistic	Average Difference	Standard Deviation	t-ratio
ψ_{links}	-1.09	2.42	-0.45
ψ_{sim}	1.35	2.18	0.62
ψ_{dyads}	2.00	2.54	0.79
ψ_D	0.12	0.36	0.33
ψ_S	-0.16	0.36	-0.44
ψ_R	0.42	1.13	0.37
ψ_T	0.19	0.86	0.22
ψ_{links}	-0.70	2.18	-0.32

The notable exception is the ψ_{dyads} statistics of the attribute prediction model which showed poor convergence characteristic with the t-ratio of 0.79 . We do have to consider however that the average difference for ψ_{dyads} is 2.00, which is not a lot considering that statistic can range from 0 to 26 (*Delinquency* has 26 actors so all statistics are scaled proportional to $k = 26$, from 0 to 26). Overall, the results in Table 1 suggest that link prediction model has better convergence properties than attribute prediction model.

It is a known fact that Exponential Random Graph Models are subject to the degeneracy [72]. The ERGM's degeneracy is the observed phenomenon where the estimated parameters of the network fail to converge, to generate (or reconstruct) the original network. The most often cited examples of degeneracy are when the sampled network becomes completely saturated with links or does not have links at all. The cause for the ERGM degeneracy are not very well understood and it had became an impediment for the wider adoption of ERGM models for the Social Network Analysis. A work by [36] which is expanded and improved follow up of the original publication on tERGM [37] addresses the degeneracy of temporal ERGM's.

The complex question of good chain-mixing for Markov's processes is well studied [56]. In our experiments the number of throw away samples and total number of samples drawn from the Markov chain were derived empirically. During training we recorded the

total number of samples vs. the number of throw away samples drawn from posterior distribution and evaluated it against achieved accuracy. As a rule of thumb we set the number of throw away examples to be a half of the total number of the drawn samples.

2.2.6 Sufficiency

The often cited work [31] described the now canonical sufficient statistics for the ERGM's such as triad counts and 2-stars. Many more sufficient statistics for the non-temporal networks were published in later works [84]. The sufficient statistics for ERGM's can be derived because of the discovery made by [11], which states that the probabilities of a random graph (social network in our case) is sufficiently described by cliques of its dependence graph. Here, the dependence graph D of a network N describes the conditional dependencies between links in N. The vertices in D are links connecting nodes in N and edges are conditional dependencies between the links given the rest of the graph N. The application of the dependence graph to define sufficient statistic was applied so far to the static non-temporal networks. It is not clear how it can be used for the temporal models such is ours because most of the statistics in etERGM are calculated on two temporally adjacent networks.

One possible way to determine the sufficiency for tERGM/etERGM could be to create a composition network $\mathbb{C}$ containing all T temporal sequences. The networks within $\mathbb{C}$ could be tied to each other by creating links between nodes at time step t and their own copies in the adjacent time steps $t-1$ and $t+1$. The dependence graph on such a network could be analyzed to confirm sufficiency of tERMG's statistics or to create a new statistics. The sufficiency question for the temporal models is a separate research topic by itself such that we do not address it in this work, however this direction is very worthy future investigation.

2.3 Experiments

To evaluate the proposed method, we conducted experiments on two synthetic datasets and two well studied real life datasets.

2.3.1 Synthetic Datasets

The purpose of experiments on synthetic data was to investigate the proposed model under controlled conditions. The synthetic datasets were generated by applying Markovian process, i.e. each generated network (time step) was derived from the previous network. To derive the next network from the previous one, we randomly selected 50% of the links in the previous network and reversed their direction. We proceeded by randomly selecting 10% of the links in the resulting network and deleting them. We counted the number of the links we have just deleted and randomly added the same number of the links to the actors pairs that were not connected. Next, we identified all incomplete transitive relationships in the network , i.e. relationships where links were present from actor p to actor q and from actor q to actor r, but link from actor r to actor p was absent. The 20% of those identified incomplete transitive relationships were randomly picked and we completed their transitivity by adding link from actor r to actor p. To keep the density of the generated network steady, we counted the number of the transitive links we just have added and randomly deleted the same number of the links from resulting network. This completed generation of one time step. The consequent network was generated by going through the same procedure, and we used the network we have just created as the starting point. The first network in the sequence was derived from the random graph. The data values of the actors were generated after the network time series were completed. Similarly to how the sequence of the networks was generated, the values of the actors from the previous time step were used to populate the next time step. To transform the k-length vector of actor attributes, where k is the number of actors, from the previous time step into the next, we randomly picked 30% of the actors from the previous time step and added to their values zero mean one standard

deviation random Gaussian noise. Additionally, the fully linked actor pairs were identified, i.e. where the link was present from actor i to actor j and from actor j to actor i, and for these actors we set the value of one actor equal to the value of its neighbor plus small random Gaussian noise. This procedure completed the generation of actor attributes for one time step. The consecutive time step was generated by applying the same procedure, where the actor attributes vector we just created was used as the starting point. To generate the first time step k-length vector of actor attributes we started off from the random Gaussian vector with zero mean and standard deviation one.

We created two synthetic datasets by following our Markovian procedure. *Dataset1* consists of 30 networks (average network density was 30%), each network consisting of 30 actors. Here, density is defined as the proportion of links to the possible number of links. *Dataset2* is a time series graph of 100 actors observed over 30 time steps. The average network density in *Dataset2* is 10%. The purpose of our synthetic experiments was to investigate the prediction accuracy of actors attributes and test our conjecture that the proposed model, which in interlocking fashion learns network structure and actors values, is superior in link prediction task to the models that only use network topology and/or time-series information. To test this assumption, we compared our algorithm with algorithms of *Common Neighbor* (CN), *Preferential Attachment* (PA), *Adamic-Adar* (AA), *Katz* (KZ), *ARIMA* (AR), *Hybrid Time Series Link Prediction Algorithm* (HA), *Tensor Factorization* (TF) and *Temporal Exponential Random Graph Model* (tERGM). For HA we used *ARIMA* and *Katz* as its input algorithms, because it was reported that these combined predictors achieve the highest accuracy [46]. For *Tensor Factorization* we used the number of decompositions in increments of 10: $L \in \{10, 20 \dots k\}$. We also compared our algorithm to tERGM, which uses a similar exponential model framework as our model, but does not consider the actors' attributes whereas our approach does. Also, a baseline "$T-1$" approach was used which assumes N^{T+1} is network N^T. As we shall see, the "$T-1$" algorithm in many circumstances performs quite well.

To measure link prediction performance we used the *area under curve* (AUC). The AUC is a preferred way to measure performance of the link prediction algorithms because social networks usually have a low link density (as low as 0.1%), which makes these datasets imbalanced [25, 46]. The output of an CN, PA, KZ, AR, HA and TF algorithms is a score matrix S, where each entry $S(i, j)$ assigns the link occurrence score proportional to the predicted probability of the link from actor i to actor j at time step $T + 1$. Our inference algorithm can also produce such score matrix. At the line 4 of our inference procedure (Chapter) we can add the sampled matrices which will result in score matrix S: $S = \sum_{k=i}^{B} \hat{N}^i$. To derive AUC from score matrix S, we moved the threshold parameter in small increments from matrix's smallest to its largest value. Each time we incremented the threshold, we created the intermediate binary matrix and set its entries for all i and j where $S(i, j) < threshold$ to 0 and the rest of the entries to 1. Thus, initially, our binary prediction matrix contained all 1's and at the end it contained all 0's. While continuously moving our threshold parameter, we recorded the percentage of *true positive* links , i.e the number of correctly predicted links divided by the total number of positive links in the target matrix, and percentage of *false positive* links, i.e. the number of predicted links that were not in the target matrix divided by the total number of negative links. The *Receive Operating Characteristics* (ROC) curve [15] is constructed by using the x-axis for false positive percentages and the y-axis for true positive ones. We calculated AUC, bounded between 0 and 1, based on generated ROC. A perfect algorithm will have AUC=1, whereas a random algorithm will have AUC=0.5. A better predictor will always have larger AUC. A similar procedure was used to obtain the prediction score matrix S from tERGM.

We calculated the mean square error (MSE) between the predicted attribute vector and true attribute vector to measure the accuracy of actor attributes prediction. The proposed method was compared with three baseline methods. One baseline method assumes that the actor attributes do not change between time steps T and $T + 1$, i.e., $\boldsymbol{x}^{T+1} = \boldsymbol{x}^T$. The second baseline was to use the history mean as the prediction, i.e., $\boldsymbol{x}^{T+1} = mean(\boldsymbol{x}^1 : \boldsymbol{x}^T)$.

Although these two baseline predictors are simple they are difficult to beat in practice. Our third baseline was the *Tensor Factorization* technique. We found that low values of L (number of decompositions) are preferable when using *Tensor Factorization* to predict attributes and in our experiments we set $L = 3$.

We also investigated how the length of the historical data used for training influences our predictors. We compared predictors by training on 1, 2, 5 and 10 previous time steps by setting up sliding windows of 1, 2, 5 and 10 training time steps. For example, in the experiments with 5 training time steps we trained predictors on time steps from 6 to 10 to predict time step 11, from 7 to 11 to predict time step 12 and so on until we predicted 30th time step. Because we only did predictions for the time steps from 11 to 30 we collected 20 AUC and MSE values for each experiment. The AUC average and sample standard deviation of each predictor on synthetic *Dataset1* are presented in Table 2. We report the MSE averages and sample standard deviation of each predictor actor attribute predictor on synthetic *Dataset1* in Table 3.

The t-test pairwise statistics, comparing etERGM results with that of the second best predictor, for each set of experiments reported in Table 2 and Table 3 were statistically significant with p-values less than 0.01.

Some of the entries in Table 2 are not filled in. We did not conduct *ARIMA* experiments using 1 and 2 training time steps because such short time-series do not provide enough data points for meaningful time-series predictions. The corresponding entries for the hybrid algorithm (HA) are not filled in either, because it uses an *ARIMA* score matrix as its input. For the *Tensor Factorization* baseline we only considered tensors with at least two time steps.

Our model requires at least one transition step for training, i.e. 2 time steps, that is why the first column in etERGM row in Table 2 and Table 3 is absent. All entries for the "$T - 1$" algorithm are the same, because no matter how many time steps used for training, "$T - 1$" needs just one previous time step to make prediction. Results presented in

Table 2: Links prediction on synthetic *Dataset1*: AUC averages and sample standard deviations of 20 experiments.

Predictor	Number of training time steps			
	1	2	5	10
AR	-	-	0.58±0.02	0.55±0.03
PA	0.62±0.03	0.61±0.03	0.60±0.02	0.58±0.03
T-1	0.71±0.02	0.71±0.02	0.71±0.02	0.71±0.02
CN	0.53±0.03	0.53±0.03	0.54±0.03	0.55±0.03
AA	0.51±0.07	0.52±0.03	0.54±0.03	0.54±0.04
KZ	0.71±0.03	0.73±0.03	0.68±0.03	0.65±0.02
HA	-	-	0.69±0.03	0.65±0.02
TF	-	0.75±0.03	0.75±0.02	0.73±0.02
tERGM	-	0.78±0.03	0.83±0.02	0.84±0.01
etERGM	-	**0.83±0.01**	**0.87±0.01**	**0.88±0.01**

Table 3: Attributes prediction on synthetic *Dataset1*: MSE averages and sample standard deviations of 20 experiments.

Predictor	Number of training time steps			
	1	2	5	10
Previous Network	1.29±0.04	1.29±0.04	1.29±0.04	1.29±0.04
Average	1.29±0.04	0.79±0.04	1.01±0.04	1.12±0.02
TF	-	0.75±0.45	1.02±0.42	1.15±0.26
etERGM	-	**0.74±0.03**	**0.70±0.04**	**0.67±0.03**

Table 2 are in many ways similar to results reported at [46] on real life data. Just like in [46] we see that when predicting from a short history (2 previous networks) *Katz* performs exceptionally well, and when it is used in conjunction with *ARIMA* as a hybrid link prediction algorithm (HA), the result is even better. We also noticed close correlations between *Common Neighbor* and *Adamic-Adar* , which were also reported in [46]. The *Tensor Factorization* algorithm also performs very well. We can see from Table 2 that in the link prediction task our model outperformed models that do not exploit actor attributes. etERGM performed better as we added more historical data to the training dataset, whereas time-invariant algorithms show the reverse trend. This observation makes sense, as under Markovian assumption our model will perform better given more historical steps to train, whereas for the time-invariant algorithms the excess of historical data will appear as noise. The results for attributes predictions reported in Table 3 are somewhat similar to the link prediction results. The only exception is *Tensor Factorization* which shows great variance predicting 20 time steps, its variance is decreasing however as the volume of training data increases. We can see that etERGM accuracy is getting better as more history was provided for training. Also, the Markovian nature of the dataset is evident by poor accuracy of the "Average" predictor as compared to the "Previous Network".

We repeated the same set of experiments on *Dataset2*. This dataset also consists of 30 time steps, with networks of 100 actors. The results of these experiments are reported in Table 4 and Table 5.

Here we observed trends similar to experiments on *Dataset1*. The pairwise t-test comparison showed that etERGM accuracy improvement are statistically significant compared to that of the second best predictor (p-values were less than 0.01).

2.3.2 Real Life Datasets

We have also conducted experiments on two real life datasets. The first, *Delinquency*, is a well studied dataset [81] consisting of 4 temporal observation of 26 students in a Dutch

Table 4: Links prediction on synthetic *Dataset2*: AUC averages and sample standard deviations of 20 experiments.

Predictor	Number of training time steps			
	1	2	5	10
AR	-	-	0.66±0.03	0.64±0.03
PA	0.68±0.04	0.66±0.04	0.67±0.04	0.64±0.04
T-1	0.71±0.01	0.71±0.01	0.71±0.01	0.71±0.01
CN	0.62±0.05	0.59±0.05	0.64±0.05	0.57±0.04
AA	0.62±0.05	0.58±0.07	0.62±0.06	0.58±0.03
KZ	0.75±0.03	0.73±0.03	0.75±0.02	0.70±0.03
HA	-	-	0.76±0.03	0.71±0.03
TF	-	0.78±0.02	0.80±0.01	0.79±0.01
tERGM	-	0.79±0.02	0.81±0.03	0.82±0.03
etERGM	-	**0.86±0.01**	**0.87±0.01**	**0.88±0.01**

Table 5: Attributes prediction on synthetic *Dataset2*: MSE averages and sample standard deviations of 20 experiments.

Predictor	Number of training time steps			
	1	2	5	10
Previous Network	0.82±0.03	0.82±0.03	0.82±0.03	0.82±0.03
Average	0.82±0.03	0.35±0.01	0.46±0.01	0.57±0.02
TF	-	0.35±0.15	0.46±0.15	0.58±0.11
etERGM	-	**0.30±0.01**	**0.29±0.01**	**0.27±0.03**

school class. For each observation, the researches asked each student to identify up to 12 pupils as friends. Also, the researchers collected delinquency measures every time the questioning was done. Delinquency measure is a five-point scale score ranging from 1 to 5, defined as a rounded average of stealing, vandalizing, fighting, and graffiti. The delinquency score 1-5 was assigned based on frequency of incidents over the last three months where :1 = never, 2 = once, 3 = 2-4 times, 4 = 5-10 times, 5 = more than 10 times. The distribution of the delinquency varied between each measurement and was highly skewed. The density of the network formed using student relationships was low (on average between 13% and 17%). The objective was to predict the relationship network and delinquency score of each student at the observation time step $t = 4$, based only on previous three surveys.

The second real life dataset used in our experiments is called *Teenagers* [61, 70]. This dataset consists of three temporal observations of 50 teenagers. Just like in *Delinquency*, teenagers were asked to identify their 12 best friends. The other measurement was the student's alcohol consumption. This measurement was defined on a 5 points scale: 1=none, 2=once or twice a year, 3=once a month, 4=once a week and 5=more than once a week. The goal in this dataset was to predict the relationship graph and the teenager's alcohol consumption score at the observation time step $t = 3$, based on two previous observations. The *Teenagers*' network density is even lower than in *Delinquency*, and is hovering at about 4.5%.

To learn the model, we have used parameters specified in Table 6 for *Delinquency* data and Table 7 for *Teenagers*. Once parameters were obtained, we used our inference technique (Chapter) to obtain predictions reported at Table 8 and Table 9. Table 8 contains the accuracies of each baseline prediction model and our proposed approach measured by AUC. In Table 9 we report the average and sample standard deviation for TF, tERGM and etERGM predictors based on 20 runs per each experiment. We can see that our approach had achieved the higher accuracy in links prediction than any other predictor on

Table 6: Parameters of *Delinquency* dataset.

Parameter	Parameter Value	Description
K	26	Number of actors
B	1000	Number of network samples
C	10000	Number of data samples
T	4	Number of time steps
σ	0.3	Similarity threshold

Table 7: Parameters of *Teenagers* dataset.

Parameter	Parameter Value	Description
K	50	Number of actors
B	1000	Number of network samples
C	10000	Number of data samples
T	3	Number of time steps
σ	0.3	Similarity threshold

Table 8: Links prediction on *Delinquency* and *Teenagers* datasets: AUC averages and sample standard deviations of 20 experiments.

	Link Prediction AUC							
	PA	T-1	CN	AA	KZ	TF	tERGM	etERGM
Delinquency	0.68	0.76	0.68	0.68	0.76	0.83±0.00	0.82±0.00	**0.84±0.00**
Teenagers	0.62	0.76	0.60	0.61	0.69	**0.84±0.00**	0.83±0.00	**0.84±0.00**

Table 9: Attributes prediction on *Delinquency* and *Teenagers* datasets: MSE averages and sample standard deviations of 20 experiments.

	Actors attributes (MSE)				
	Previous Network	Average TF		etERGM	p-value
Delinquency	1.12	1.01	1.00±0.00	**0.94±0.02**	$p < 0.05$
Teenagers	0.90	0.87	0.87±0.00	**0.83±0.01**	$p < 0.05$

Delinquency. The pairwise t-test comparison showed that etERGM accuracy improvement are statistically significant compared to that of the *Tensor Factorization* - the second best predictor (p-values were less than 0.01). For the *Teenagers* dataset, the results of TF and etERGM were the same. Table 9 contains the averages, sample standard deviation and analysis of statistical significance of the actor attributes predictions on both real life datasets, which are based on the same 20 runs. Here, the proposed approach, as measured by MSE, outperforms baseline predictors. The *Tensor Factorization* algorithm performs very well in link prediction task, however for the attribute prediction its results seem to correlate with *Average* predictor.

Overall, in both experiments on real life datasets the etERGM outperformed the conventional predictors in prediction of actor's attributes and the link prediction in most cases.

2.4 Scalability

Empirical observations suggests that our approach is scalable to handle networks with hundreds of actors. The largest network size we conducted our experiments on had 500 actors and it is possible to handle a number slightly larger than that. This is sizable improvement compared to the htERGM model [35], which we consider to be closely related to ours. htERGM can only handle networks of up to 10 nodes (actors). The increase in efficiency over htERGM was achieved by decoupling link prediction and attribute prediction probabilistic models, whereas htERGM's model was joint.

Our approach is not directly scalable to the networks the size of Youtube or Facebook. This is attributable to the fact that most of our statistics/features have runtime of $O(n^2)$. Also, the features are constantly recalculated as part of the Gibbs sampling, which is inherently slow. However, it has been noted that networks of such size are impractical for the problem we are addressing here, because humans are incapable of handling more than couple hundred relations simultaneously. Consider a class with 500 students where researchers conducted multiple temporal observations of social relationships by asking each student to

identify each student's friends. We know that any given student will indicate an absence of a link to the majority of other students. The links will be absent not because of the student's personal dislike of the students, but simply because she never had a chance to get to know so many pupils. The similar conclusion that social networks with an invariant set of actors should not exceed a couple hundred nodes can be found in [81]. One possible way to scale up our approach to handle large networks is to run a community detection algorithm which would find the network clusters containing up to a few hundred nodes. The etERGM then can be applied on detected communities separately to make predictions. Such an approach would make it unnecessary for etERGM to consider all $n^2 - n$ relationships, which is a valid assumption because we know that people in large networks (such as Facebook) are only aware of the people within or close to their social circle. How to scale the etERGM to the large networks is out of scope of this work, but we will try to address it in our future research.

In recently published work on tERGM [36], our predecessor, authors discovered that with a smart choice of statistics for link transition probabilities it is possible to completely avoid costly parameter estimation procedure described in the previous chapters. In a special case the choice of the tractable ψ function makes it unnecessary to perform expensive sampling steps, because it becomes possible to do exact Newton's updates. We leave investigation of such statistics and their empirical effects on the prediction accuracy for future research.

We have investigated the runtime behavior of our approach on two sets of experiments. In the first experiment we measured how the length of the historical data used for training influences the runtime of our approach. We achieved this by running our algorithm on the synthetic dataset with 30 actors. For each run we changed the number of time steps used for training and we recorded the time the algorithm took to learn the parameters of both models and to do the inference. Results of this experiment are shown in Figure 2.

We can see that there is an obvious linear trend between the runtime and the amount

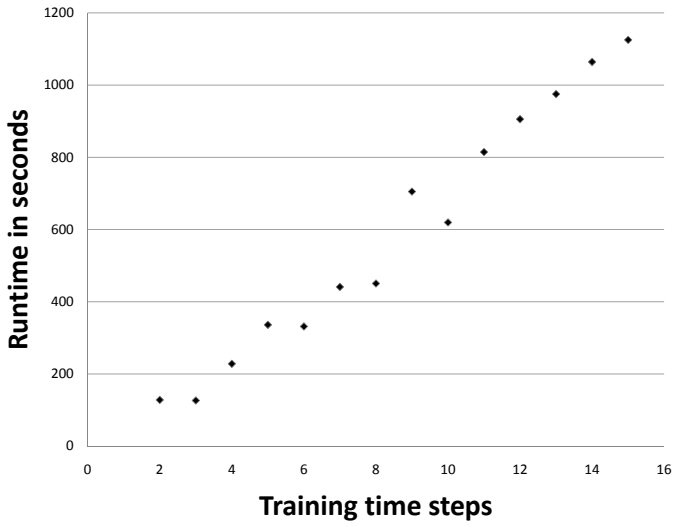

Figure 2: etERGM runtime in seconds vs. the number of time steps used for training for network with 30 actors

of historical data used for training. In the second experiment we used the synthetic dataset with four previous networks, where three were used for training and one for prediction.

In this experiment we were gradually changing the number of actors from 30 to 100 in increments of 5 and for each change we recorded the time it took for the algorithm to run. The results of the second experiment are reported in Figure 3. Here we observe the quadratic trend between the runtime and the network size. This is expected, because as the number of actors grows we considered the quadratic increase in the number of possible relationships.

Despite the network size limitation, there are real life problems that could benefit from our approach. Besides the two real life datasets presented here, there are temporal network of similar size in sociological and biological domain.

2.5 Discussion

We have shown that the log linear etERGM is a viable predictor for links and attributes in temporal social networks. One of its core strengths is the separate learning of its two

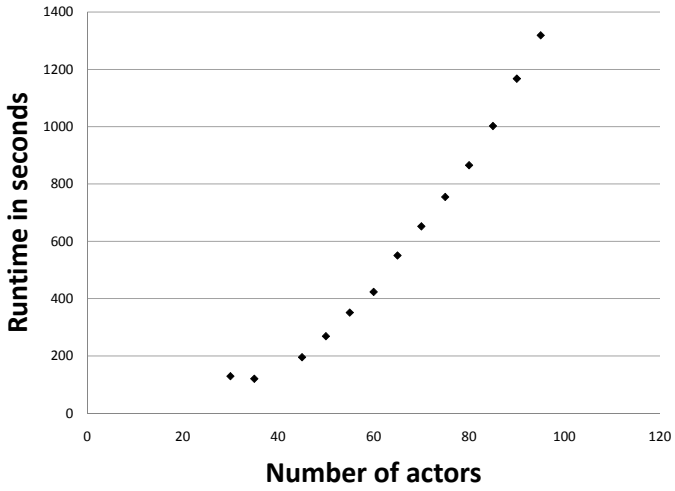

Figure 3: etERGM runtime in seconds vs. the number of actors in the network (3 time steps used for training)

component models, which makes it applicable to larger problems. In our study, only actors with a single real-valued attribute were considered. In many settings, more information is collected about participating actors. As a more general task the model perhaps could be adapted to include multivariate attributes. Despite the fact that we specifically conducted the testing on temporal social networks, it would be also interesting to find out how the new method would perform on gene expression networks.

CHAPTER 3

IMPUTATION FOR LONGITUDINAL SOCIAL SURVEYS

A special case of temporal social networks reviewed in previous chapter are longitudinal social surveys. Such social surveys suffer notoriously from underreported and missing data. In this chapter we demonstrate how modification of log linear model etERGM, described earlier in Chapter , can be applied for imputation of links and actor's attributes in social networks/surveys. Previous research done in social surveys domain had shown that social surveys are especially vulnerable to missing data. It was shown that there are very few circumstances under which researchers might be able to use a social network with missing data. A study by [54] had shown how various missing data mechanisms (network boundary, survey non-response and vertex degree censoring) dramatically affect estimation of network statistics. The statistical simulation experiments done in [54] suggest that actors' non-response greatly underestimates clustering and assortativity coefficients [20] leading to inflated measurement errors.

The vast majority of the published work investigates the effects of various imputation techniques on SNA [48]. Works by [53, 74] sought to provide an accurate estimation of network statistics. In [74], the exponential random graph p^* model (ERGM) [31], reviewed in Chapter , was developed for recovery of the network measurements. Study on how imputation techniques affect network statistics and their estimation is a separate research topic which is not addressed here because we are more interested in comparing the imputation accuracies.

The practical benefits of social network imputation were illustrated in the study of criminal gangs in the city of Los Angeles [87]. The problem formulated in [87] is different than the problem addressed in this chapter, however, we discuss it to underscore the importance

of the research in this field. The problem faced by Los Angeles law enforcement was to identify which gang affiliates were involved in criminal activities such as murder, drive-by shooting etc. There were twenty-nine active gangs in Los Angeles at the time of the study. For the most part, police knew which gangs were involved in the fights. However, in some incidents the only gang affiliation available was the one of the victim. The problem addressed in [87] was to probabilistically identify (impute) the other gang which participated in confrontation.

We consider the case of the non-responsiveness by surveyed actors. Due to various reasons certain actors at different observation times might choose to ignore the questionnaire provided to them by social scientists. The non-respondent might choose to ignore the questions because of the personal reasons. The panel mortality, where people drop out from the longitudinal survey and cannot be located, is also a possibility. Once the person ignores the questionnaire or drops from the study, she might reappear in the future wave panel(s). For example, if survey panel observations were done at times $t = 1 \dots 4$ we can have actors who completely ignored all four wave panels, we can also have actors fully participating in all of the surveys. We might encounter the situation when actors have chosen to respond to any combination of the wave panels. In our study we assume the real valued attribute of the non-respondent actor (alcohol usage score, for example) is also not known. Most importantly, the actor who ignored the wave panel is never completely unobserved, because the other participants might indicate a friendship link to her.

More formally, given the sequence of the wave panels surveyed at times $t = 1 \dots T$ denoted as $N^1 \dots N^T$, the corresponding actors attributes denoted as $\mathbf{x}^1 \dots \mathbf{x}^T$ ($\mathbf{x}^t$ is $1 \times k$ real valued vector, k is the number of actors), and the unobserved actors sets $S^1 \dots S^T$ ($|S^t| = m^t$, $0 \leq m^t < k \quad \forall t$) our goal is to impute the outgoing links and real valued attributes of the non-respondent actors from the set S^t for each time step t.

3.1 Related Work

The treatment of the unobserved links in social networks has been an active area of research for many years. Most of the published work investigated the relationship between the imputation techniques and the introduction of statistical bias and loss of statistical power in a single stationary network [47], or in the context of longitudinal social networks [83]. Here, we rely in the set up of our experiments on one of these works [48] because it provided a nice comprehensive foundation on how to treat a missing data in longitudinal networks. The question of quality of the recovered network statistics is very important. However, we are more interested in accuracy of the imputation techniques. In this chapter, we cover some of the most common and recent imputation methods of links and actor's attributes which will serve as baselines in our experiments.

3.1.1 Link Imputation Techniques

Reconstruction: A simple but powerful approach to reconstruct links in a single stationary network was proposed at [88]. This reconstruction method takes advantage of the reciprocity effect very often found in social networks. The imputation procedure works as following: for all ties between respondent and non-respondent actors impute the unobserved link as opposite to the one that was observed: $N^t_{ij(imputed)} = N^t_{ji(observed)}$. For ties between the non-respondents impute the link with random probability of the observed network density. Here we define the network density as $d = \frac{\sum_{ij}^{k} N^t_{ij}}{k(k-1)}$.

Preferential Attachment: The Preferential Attachment technique proposed at [8] is based on the assumption that actors with many social links are more likely to be connected to each other. This technique postulates that the probability of missing actor i having a link to actor j (observed or unobserved) is proportional to indegree r_j of actor j: $P(r_j) = \frac{r_j}{\sum_{i \neq j} r_j}$. i.e. a "popular" actor is more likely to have an incoming link from a missing actor. In the Preferential Attachment procedure, for each unobserved actor i we randomly draw the outdegree number q_i from outdegree distribution of the observed net-

work. For the same missing actor we randomly draw, without replacement, and according to probability $P(r_j)$, the q_i number of actors (observed or unobserved). In this step, actors who are popular are more likely to be selected than less popular actors (actors with less incoming links). Finally, we impute the links from actor i to actors which were selected as $N^t_{ij(imputed)} = 1$ and $N^t_{ij(imputed)} = 0$ to the ones that were not selected.

Constrained Random Dot Product Graph The Constrained Random Dot Product Graph (CRDPG) [60] is an imputation technique which models actors as residing in s-dimensional latent space. In this model, the dot product of actors' pair latent coordinates yields the probability of the link between the two:

$$p_{ij} = f(\mathbf{x}_i \cdot \mathbf{x}_j + \tilde{\mathbf{y}}_i \cdot \tilde{\mathbf{y}}_j) \tag{35}$$

In Equation (35), f is a simple threshold function

$$f(x) = \begin{cases} 0, & x \leqslant 0, \\ x, & 0 \leqslant x \leqslant 1, \\ 1, & x \geqslant 1. \end{cases} \tag{36}$$

and $\mathbf{x}_i, \mathbf{x}_j$ are $\mathbf{x} \in \mathbb{R}^d$ latent coordinate vectors of actors i and j in d dimensional latent space. Finally, $\tilde{\mathbf{y}}_i = \mathbf{y}_i * \alpha_i$, where $\mathbf{y}_i$ is actor's i covariates vector and α_i is its associated weights, $*$ is component-wise multiplication. The learning of model parameters is done via the iterative maximum likelihood estimation algorithm described in [60].

Multiplicative Latent Factor Model: The approach by Hoff models the links' prediction in terms of logistic regression with an extra latent variable [43]. In [43] the probability of the link is expressed by:

$$\log \text{odds}(y_{i,j} = 1) = \boldsymbol{\beta}' \boldsymbol{x}_{i,j} + z_{i,j} \tag{37}$$

Here, β is the vector of logistic regression coefficients and $\mathbf{x}_{i,j}$ are known predictor

variables of the relationship pairs. The latent variable $z_{i,j}$ represents patterns in the data unrelated to known predictors. Hoff proposed to use random matrix $\mathbf{Z}$ of latent effects set to deviations of the log-odds from the linear predictor $\beta' \mathbf{x}_{i,j}$. The random matrix $\mathbf{Z}$ is composed of mean matrix $\mathbf{M}$ and noise matrix $\mathbf{E}$: $\mathbf{Z} = \mathbf{M} + \mathbf{E}$. The mean matrix $\mathbf{M}$ of systematic effects is further decomposed using singular value decomposition into lower lank approximation. Such decomposition allows for a better representation of main data patterns and eliminates the lower-order noise. The multiplicative latent factor model is suitable for link imputation, it is proposed in [43] to train the model on the observed part of a dataset and then use model parameters to impute unobserved responses.

Random: We added Random imputation to our baselines more as a sanity check than as a serious predictor. This procedure will randomly fill-in the unobserved portion of the sociomatrix according to the random probability of the density d of the observed part.

None of the above-mentioned link imputation techniques consider the real-valued attributes of the observed actors. These methods also do not take advantage of the temporal nature of the longitudinal social survey as they can only impute one stationary network at a time without consideration of other networks in the temporal sequence. Our technique, which we discuss in Chapter , will bridge this gap.

3.1.2 Actor's Attribute Imputation Techniques

In our study we assume the non-respondent actors also fail to provide any other personal information sought by researchers. If the missing information is not available "a priori", then it also has to be imputed. We will only consider the case of a single real valued attribute per each actor per one time step because our goal was to evaluate our approach on a simpler model. Also, the multivariate datasets are somewhat hard to obtain.

Here we will discuss two imputation techniques for missing real valued actor attributes which will serve as baselines in our experiments.

Average: The Average method imputes the missing actor's attribute value at each time

step as the average of the observed actors in the same survey. This technique is simple and crude, but sometimes simple methods can provide good results.

DynaMMo: The DynaMMo algorithm proposed at [57] is specifically designed to impute the information gaps in multivariate temporal sequence data. The real valued actor attributes in our problem, where each temporal observation is a k-dimensional multivariate variable $\mathbf{x^t}$ (k is the number of actors), is in effect such a multivariate temporal sequence which can be imputed by DynaMMo without any modifications. The probabilistic model of DynaMMo consists of two multivariate Gaussian processes. First process models the transition probabilities between the time steps in the multivariate latent space: $\mathbf{z}_{n+1} = \mathbf{F}\mathbf{z}_n + \omega_n$. The second process describes emission from the latent space to the observed: $\mathbf{x}_n = \mathbf{G}\mathbf{z} + \epsilon_n$. Here $\mathbf{F}$ is transition and $\mathbf{G}$ is observation projections and ω_i, ϵi are multivariate Gaussian noises. This model is similar to Linear Dynamical System except it includes an indicator matrix $\mathbf{W}$ of missing values. The joint distribution of observed values $\mathbf{X}_m$, unobserved values $\mathbf{X}_g$ and latent space $\mathbf{Z}$ is expressed as:

$$P(\mathbf{X}_m, \mathbf{X}_g, \mathbf{Z}) = P(\mathbf{z}_1) \cdot \prod_{i=2}^{T} P(\mathbf{z}_i|\mathbf{z}_{i-1}) \cdot \prod_{i=1}^{T} P(\mathbf{x}_i|\mathbf{z}_i) \tag{38}$$

where T denotes the number of time steps. In Equation (38), the learning of latent parameters $\mathbf{Z}$ is done through iterative coordinate gradient descent optimization procedure. After the model parameters are learned, the imputation of missing values is easily computed from estimation of the latent variables and using the Markov property of the model.

In the next chapter we show how we have incorporated log linear etERGM's prediction models (Chapter) into our solution for imputation of longitudinal social surveys. etERGM is a natural fit for the problem we are addressing here because it considers the temporal nature of the surveys and interdependence of actors' links and attributes (homophily selection). We will show how etERGM characteristics allow us to build our own state-of-the-art log linear imputation technique.

3.1.3 Other Relevant Works

Recent advancements [25, 94] in tensor decomposition led to a number of interesting models that are relevant to our work. In [94] it was proposed to extend probabilistic matrix factorization [77] for the task of predicting user's recommendations in temporal settings. The model in [94] predicts a real valued recommendation score R_{ij}^t by user i of a merchandise j at the time t. The score is modeled as Gaussian distribution with mean set to outer product of three latent vector coordinates (one vector coordinates corresponds to user, the second vector models merchandise and third describes time axis). This model, in principle, could be adapted to impute social links' however there are major difficulties it would need to overcome to be suitable for our problem: its output is a real valued score and we are interested in the probability of the link, and it does not predict nor include information about nodes.

3.2 The Proposed ITERGM Approach

Before we discuss our approach we should explain the Area Under the Curve (AUC) measure and how it is computed to measure link prediction accuracy. Readers familiar with the use of AUC for link prediction in social networks can safely skip next paragraph.

In general, to measure the link prediction accuracy of a temporal network sequence the AUC was used successfully in the past [25, 46]. The AUC is a preferable measurement in the presence of imbalanced datasets such as social networks where link density is usually low. Every link imputation algorithm covered here is non-deterministic. Therefore one possible way to measure the link imputation accuracy on a single social network is to compute a score matrix $\mathbb{S}$:

$$\mathbb{S} = \sum_{l} N_l \tag{39}$$

Each run of an imputation algorithm results in the binary $|S| \times k$ subset matrix N_l, which contains only imputed outgoing links. Here, S is the set of actors who did not respond

to survey (did not indicate their outgoing links) and k is a total number of actors in the network. Thus the resulting score matrix $\mathbb{S}$ contains the probabilities scores of all imputed links. Using such a matrix we can construct a Receiver Optimization Curve (ROC) by moving the *threshold* parameter in small increments from the matrix's $\mathbb{S}$ smallest to its largest value. Each time we move the *threshold* we create an intermediate binary matrix and set all its entries to 0 if $\mathbb{S}(i,j) < threshold \; \forall i,j$ and 1 otherwise. Therefore, a binary prediction matrix at the beginning contains all 1s and it contains 0s at the end. While moving the *threshold* parameter we calculate the true positive and false positive rates of imputed links against the true target. True positive rate is number of correctly imputed links divided by the total count of true links. False positive rate is number of imputed links which were not in the true target divided by the total count of non-existing links (structural zeros). We construct ROC by using the x-axis for the false positive rate and the y-axis for the true positive. We calculate AUC, bounded between 0 and 1, based on the constructed curve. A perfect imputation algorithm will have AUC=1, and random algorithm will have AUC=0.5. A better predictor always have larger AUC.

The imputation methods we have reviewed so far (Chapter and) can either be applied for link or attribute prediction, or completely ignore the temporal aspect of the surveys. The etERGM model (Chapter) provides many properties we are looking for in our imputation approach: it encodes the interdependence of actors attributes and links, it also considers the time axis in its learning and inference process. Despite all its characteristics, the etERGM cannot be applied directly to impute attributes or links. Its probability models, Equations (21) and (25), can only predict the social network structure at the next unobserved time step given all completely observed previous time steps. New algorithm, named ITERGM, is in essence the Expectation Maximization (EM) algorithm over two Markov Chain Monte Carlo (MCMC) inferences. During Expectation step we draw multiple particles from both link and prediction models (Steps 4 and 6) of etERGM and in interlocking fashion use them to impute/update the dataset. During Maximization (Steps 3 and 5) we relearn both models'

parameters on the updated data. We repeat these steps until the weights of both models have converged. We choose the iterative solution over a single pass because we want to avoid the dependency of the imputation results on the initialized values. It is unlikely that a single update/imputation pass would reach the point of maximum likelihood. Therefore, we re-learn the model parameters via an iterative approach. More formally, ITERGM method consists of the following steps:

Algorithm 1 ITERGM

Input: The sequence of surveys:$N^{1:T}$, $\mathbf{x}^{1:T}$ where links and attributes, corresponding to actor sets $S^{1:T}$ are unobserved: $\mathbf{x}^t(S^t) = \varnothing$ and $N^t(S^t, j) = \varnothing \ \ \forall t, j$

Output: Imputed links score matrices: $\mathbb{S}^{1:T}$. Imputed actors' attributes: $\mathbf{x}^{1:T}_{imputed}$

1: Initialize iteration counter: $iter = 1$
2: Apply DynaMMo (Chapter) to initialize missing values in $\mathbf{x}^{1:T} \rightarrow \mathbf{x}^{1:T}_{temporary}$
3: **for** t in $1 \ldots T$ **do**
4: Impute $N^t(S^t, j), \forall j$ with best link imputation technique from Chapter $\rightarrow$ $N^t_{temporary}$
5: **end for**
6: Train etERGM's attribute prediction model (Chapter) on $N^{1:T}_{temporary}$, $\mathbf{x}^{1:T}_{temporary}$ to learn weights $\boldsymbol{\gamma}_{iter}$
7: **for** t in $2 \ldots T$ **do**
8: Sample multiple vectors $\mathbf{x}^t_{inferred}$ from distribution $P(\bar{\mathbf{x}}^t_{inferred} | \mathbf{x}^{t-1}_{temporary}, N^t_{temporary}, \boldsymbol{\gamma}_{iter})$
9: **for all** missing actor p in S^t **do**
10: $\mathbf{x}^t_{temporary}(p) = mean(\mathbf{x}^t_{inferred}(p))$
11: **end for**
12: **end for**
13: Train etERGM's link prediction model (Chapter) on $N^{1:T}_{temporary}$, $\mathbf{x}^{1:T}_{temporary}$ to learn weights $\boldsymbol{\theta}_{iter}$.
14: **for** t in $2 \ldots T$ **do**
15: Draw multiple networks $N^t_{inferred}$ from posterior distribution:$P(\bar{N}^t_{inferred} | N^{t-1}_{temporary}, \mathbf{x}^t_{temporary}, \boldsymbol{\theta}_{iter})$.
16: Calculate $|S^t| \times k$ score matrix: $\mathbb{S}^t = \sum N^t_{inferred}(S^t, j), \forall j$
17: Set $N^t_{temporary}(S^t, j) = bestcut(\mathbb{S}^t), \forall j$
18: **end for**
19: **if** $\boldsymbol{\theta}_{iter}$, $\boldsymbol{\theta}_{iter-1}$ and $\boldsymbol{\gamma}_{iter}$, $\boldsymbol{\gamma}_{iter-1}$ had converged **then**
20: $\mathbf{x}^{1:T}_{imputed} = \mathbf{x}^{1:T}_{temporary}$
21: **return**
22: **else**
23: $iter = iter + 1$
24: **go to Step 6**
25: **end if**

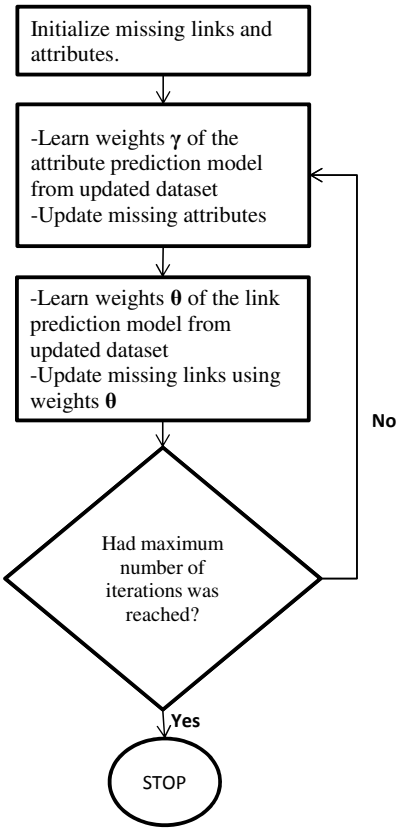

Figure 4: Schematic diagram of the proposed algorithm ITERGM.

The input of the algorithm is the temporal sequence of the partially observed sociomatrices and actors' attributes. The Steps 1-5 of the algorithm are initializations. In Step 2 we chose "a priori" the DynaMMo algorithm to initialize the missing values of the multivariate temporal sequence of actors' attributes. In Steps 3-5 we apply every imputation technique outlined in Chapter to every partially observed sociomatrix and choose the best imputation procedure for initialization of the network's unobserved part. We apply straightforward criteria to select the best initialization technique for links, we choose the algorithm in which imputed density is closest to the density of the observed part. For example, assume that link density of the observed part of network N^t is 0.2. We impute the unobserved part of network N^t by applying every algorithm described in Chapter and record the resulting link density of the unobserved part. To initialize links in N^t, we pick the algorithm with computed density of the unobserved part closest to 0.2.

At this point, all links and attributes of all networks have been initialized and we begin our iterative approach (Steps 6-25). In Steps 7-12 of the algorithm we apply the etERGM node prediction model to learn its weights and to impute the unobserved attributes by drawing samples from the model over the set of the unobserved actors. Then, in Step 13, we learn the weights of the etERGM link prediction model by training it on the dataset we have just updated with imputed actors' attributes in interlocking fashion. Knowing the weights, we draw multiple samples from the link prediction model and use them to impute the outgoing missing links (Steps 14-18). In Step 19 we check if we reached number of maximum iterations. If the number of maximum iterations is not reached, we continue the learning/inference process constantly updating the dataset over the set of unobserved actors in interlocking fashion and re-learn the weights of etERGM. Otherwise, the score matrices in Step 16 are our prediction of the imputed links and $\mathbf{x}^{1:T}_{temporary}$ is our imputed temporal sequence of actors' attributes (Step 20). We present the schematic outline of the ITERGM algorithm in Figure 4. It is important to note that at each transition we consequently update the missing part of the same dataset (links and attributes) based on the model parameters which were learned at the previous iteration.

The expectation steps of our algorithm deserve closer attention. Computing the expected values of the missing actors' attributes is fairly straightforward. In Step 8, for each survey we sample multiple particles (actors' attributes vectors) based on the weights $\boldsymbol{\gamma}_{iter}$ learned in the current iteration. We take the mean of the corresponding values of the actors' attribute vector as our prediction of the missing actor's attribute and use that to update our dataset (Steps 9-11). The inference of the links is a bit more involved. Similarly to the imputation of actors' attributes we sample multiple sociomatrices for every survey based on the present learned weights $\boldsymbol{\theta}_{iter}$ of the link prediction model (Step 15). We take the predictions of the imputed values in the form of the score matrices $\mathbb{S}^t$ by adding drawn samples in Step 16. In their present form the score matrices $\mathbb{S}^t$ cannot be used directly to impute the missing links. We have to convert $\mathbb{S}^t$, which hold the relative probability scores

of possible links, into binary form and use that to update the missing part of network (sociomatrices are always binary). That is why in Step 17 we apply the *bestcut* procedure to determine the best *threshold* or "bestcut" to make a binary link imputation matrix suitable to update missing links. The *bestcut* procedure chooses the *threshold* such that the resulting binary matrix is maximizing the probability of the link prediction model in Step 15. This is achieved by moving the *threshold* from the score matrix's smallest to its largest value. Each resulting binary imputation matrix is substituted into a link prediction model and we pick the best matrix ("cut") which maximizes the link prediction probability.

3.3 Algorithm Convergence

Our algorithm in its essence is a continuous sampling from link and attribute prediction models with iterative updates of model weights. Our exit condition is a sufficient number of iterations, which in practice we limit to 3 or 4. However, we have to ensure that our technique indeed converges. To evaluate convergence of our algorithm we adapted a standard convergence evaluation technique for ERGM models [82] previously discussed in Chapter . It works as following: a) take a fully observed network and calculate its real observed statistics ψ_0, b) randomly remove a given percentage of actors from the network and apply the imputation technique, c) during imputation draw multiple samples from the model and for each drawn sample calculate network statistics ψ_k, d) calculate the t-ratio as Equation 34. In [82] it was suggested that $|t_k| \leq 0.1$ is indicative of an excellent convergence, $0.1 < |t_k| \leq 0.2$ is good and $0.2 < |t_k| \leq 0.3$ is fair.

We evaluated the convergence property of our algorithm on the real life dataset *Delinquency* which was described in Chapter . We picked a single transition step from $t = 2$ to $t = 3$ of the dataset and removed 20% of the actors at random from the network at step $t = 3$. We then ran our imputation technique on the selected transition step and after 3 iterations had collected 1,000 samples of sociomatrices and actors' attributes. In Table 10 we present the averages of the differences between the true statistics ψ_0 and statistics based on

Table 10: Convergence estimates of the imputation algorithm on time steps $t = 2, 3$ of the real life dataset *Delinquency*

Statistic	Average Difference	Standard Deviation	t-ratio
ψ_{links}	0.74	2.12	0.35
ψ_{sim}	-0.59	2.09	-0.28
ψ_{dyads}	-0.48	1.20	-0.40
ψ_D	2.02	1.85	1.09
ψ_S	-1.82	1.98	-0.92
ψ_R	0.00	1.01	0.00
ψ_T	0.78	1.15	0.68
ψ_{links}	-0.82	1.55	-0.53

the imputed samples, the standard deviation of the differences and corresponding t-ratios. The etERGM statistics $\psi_{links}, \psi_{sim}, \psi_{dyads}, \psi_D, \psi_S, \psi_R, \psi_T$ shown in Table 10 correspond to the measurements of homophily, attributes' stability, similarity, density, links stability, reciprocity and transitivity (Chapters and).

In Table 10 we observe that converging properties are ranging from excellent to poor. However, it should be noted that none of the t-ratios indicate statistical significance. This means that network statistics derived from the imputed data are not significantly different than true values.

3.4 Experiments

To evaluate the accuracy of our approach, we conducted a series of the experiments on synthetic and real life datasets. Two approaches, Missing At Random (MAR) and Missing Not At Random (MNAR), are used to model the non-responses in social network literature [48]. The former approach assumes there is no underlying hidden structure explaining the missing information, the latter assumes that the missing values are dependent on the actors' attributes or the network topology. For both synthetic and real life datasets we set up our experiments as follows: we randomly remove a predefined percentage of the actors from each wave panel according to MAR or MNAR. We perform repeated imputations ($u = 5$) on the semi-observed dataset by applying the proposed approach and baseline imputation

techniques for links and attributes. To compare results we construct the 90% confidence intervals on both link and attribute imputations according to the "multiple imputation" technique [79]. We run our experiments by simulating the removal of the actors according to five missing mechanism: two MAR and three types of MNAR.

For the first MAR, called "Random", we removed actors at each time step completely at random. At this scenario an actor randomly removed at one time step can potentially reappear at the next time step(s). For the second MAR scenario, called "Absent", an actor was randomly removed completely from all panels. The "Absent" mechanism is useful for modeling of actors who did not respond to a single survey. Such a scenario can occur in a classroom if, for example, a student was out sick for the duration of the study or perhaps persistently ignored a survey. This scenario can be also observed in enclosed medical setting such as a hemodialysis clinic where patients are often absent from their regular environment setting due to hospitalization [24].

For MNAR, we removed the actors according to probabilities of $\frac{1}{(x_i^t)^2}$, $\frac{1}{(1+indegree)^2}$ and , $\frac{1}{(1+outdegree.)^2}$. The first MNAR, which we call "Score", models the absence of actors as being dependent on their real-valued attribute (for example, the actors with higher alcohol consumption score are less likely to respond to survey). The second MNAR, called "Indegree", assumes that the more popular actors are more likely to be survey participants. The third, called "Outdegree", assumes that the socially inactive people are less likely to be willing to answer survey questions. Then, for each missing mechanism we have removed 20%, 40% and 60% of actors from each survey. To summarize, we model the actors' removal at the five types of missingness and three different percentages, to the total of 15 sets of experiments per each dataset and we repeat imputation of each set 5 times.

To assess the imputation accuracy of the actors' attributes we used the Mean Squared Error (MSE) measurement. For the imputation of the links we calculated the Area Under Curve (AUC) measurement on the score matrix of the imputed part of the sociomatrix. A perfect imputation algorithm will have AUC=1 and a random algorithm will have

AUC=0.5, larger AUC value indicates a better algorithm.

3.4.1 Synthetic Dataset

The purpose of the experiments on the synthetic dataset is to verify the proposed imputation technique under controlled conditions. We generated one synthetic dataset adhering to the Markovian process, where each consecutive social network in the temporal sequence at time t is created from the network of the previous time step $t-1$. We started the generation process by creating a random graph, denoted as N^1 , and set $t = 1$, then repeat until the needed number of the networks is generated:

- set $t = t + 1$
- create a copy of the previous network by setting $N^t = N^{t-1}$
- randomly inverse direction of the 50% links in N^t
- randomly reassign 10% of the links in N^t
- randomly pick 20% of incomplete transitive relationships (link is present from p to q, and q to r, but not from p to r) in N^t, and complete the transitive relationships by adding closure links (from p to r)
- count the number of links added to complete transitive relationships and randomly delete the same number of links from the graph N^t

To generate the actors' attributes we used an approach similar to the network generation procedure. We started the generation of the actors' attributes after the networks generation was complete. Similarly to how the sequence of the networks were created we started from the random k-length vector of actors attributes $\boldsymbol{x}^t$ ($t = 1$) drawn from the Gaussian distribution with zero mean and one standard deviation. We repeated the following steps until the actors' attributes were populated for all the networks in the sequence:

- set $t = t + 1$

Table 11: Links and attributes imputation on the synthetic *Dataset1* of 1000 actors observed at four time steps (simulating five types of missing mechanisms for 20%-60% of missing actors): 90% confidence intervals of the AUC and MSE. **Bold** denotes the best result.

Type	%	Link Imputation AUC					
		Random	PrAtt.	Recon.	CRDPG	Mult.Lat.F.	ITERGM
random	20	0.51±0.00	0.53±0.00	0.51±0.00	0.54±0.00	0.60±0.00	**0.62±0.01**
	40	0.51±0.00	0.52±0.00	0.51±0.00	0.55±0.00	0.58±0.00	**0.63±0.01**
	60	0.49±0.00	0.52±0.00	0.51±0.00	0.54±0.00	0.58±0.00	**0.59±0.01**
absent	20	0.50±0.00	0.54±0.01	0.51±0.00	0.53±0.00	0.57±0.00	**0.59±0.01**
	40	0.51±0.02	0.52±0.02	0.52±0.00	0.53±0.00	0.54±0.00	**0.57±0.01**
	60	0.50±0.02	0.52±0.00	0.50±0.00	0.51±0.00	0.52±0.00	**0.56±0.01**
score	20	0.51±0.00	0.54±0.00	0.51±0.00	0.53±0.00	0.59±0.00	**0.64±0.02**
	40	0.50±0.00	0.54±0.00	0.50±0.00	0.55±0.00	0.57±0.00	**0.62±0.01**
	60	0.52±0.00	0.52±0.00	0.50±0.00	0.54±0.00	0.53±0.00	**0.57±0.00**
indegree	20	0.50±0.00	0.54±0.00	0.52±0.00	0.56±0.00	0.62±0.00	**0.65±0.00**
	40	0.50±0.00	0.54±0.00	0.51±0.00	0.55±0.00	0.62±0.00	**0.63±0.00**
	60	0.51±0.00	0.53±0.00	0.52±0.00	0.53±0.00	0.58±0.00	**0.61±0.00**
outdegree	20	0.51±0.00	0.53±0.00	0.52±0.00	0.54±0.00	0.59±0.00	**0.64±0.00**
	40	0.50±0.00	0.54±0.00	0.51±0.00	0.56±0.00	0.61±0.00	**0.64±0.01**
	60	0.51±0.00	0.52±0.00	0.53±0.00	0.53±0.00	0.56±0.00	**0.59±0.01**

Type	%	Attributes Imputation MSE		
		Average	DynMM	ITERGM
random	20	0.15±0.00	0.14±0.00	**0.10±0.00**
	40	0.19±0.00	0.31±0.00	**0.15±0.00**
	60	0.41±0.00	0.24±0.00	**0.18±0.00**
absent	20	0.26±0.00	0.21±0.00	**0.16±0.00**
	40	0.38±0.00	0.35±0.00	**0.22±0.01**
	60	0.49±0.00	0.38±0.00	**0.27±0.01**
score	20	0.25±0.00	0.22±0.00	**0.10±0.00**
	40	0.37±0.00	0.24±0.01	**0.16±0.01**
	60	0.60±0.00	0.42±0.00	**0.35±0.01**
indegree	20	0.45±0.00	0.15±0.00	**0.12±0.00**
	40	0.35±0.00	0.20±0.00	**0.14±0.00**
	60	0.31±0.00	0.29±0.01	**0.22±0.01**
outdegree	20	0.24±0.00	0.31±0.01	**0.20±0.01**
	40	0.21±0.00	0.22±0.00	**0.17±0.00**
	60	0.40±0.00	0.49±0.00	**0.35±0.00**

- create a copy of the previous actors' attributes by setting $\boldsymbol{x}^t = \boldsymbol{x}^{t-1}$
- randomly select 30% of actors from $\boldsymbol{x}^t$ and add to their attribute values a zero mean one standard deviation Gaussian noise
- identify all doubly linked actors pairs in N^t, set the value of one actor in each of these pairs to the value of its doubly linked counterpart plus small random Gaussian noise

We created one synthetic dataset *Dataset1*, simulating a network of 1000 actors observed at four time steps, by following our procedure. On average, it took ITERGM four iterations to achieve convergence on this synthetic dataset. The results of the experiments on this dataset are presented in Table 11. In these experiments the ITERGM had the best imputation accuracy of the actors' links and attributes as compared to the baselines every time. We also conducted similar experiments, not presented in this chapter, on synthetic networks with 30 and 500 actors. The results were analogous to the ones presented in Table 11. We did notice however that as the size of the network grows the accuracies of the imputation techniques (baselines and ITERGM) were dropping. We attribute this phenomenon to the fact that methods presented here are addressing the imputation globally and do not consider local properties of the communities/clusters that are often found in the large networks.

To further characterize our approach, we conducted experiments on eleven synthetic datasets of increasing numbers of actors ranging from 20 to 1,000. All generated datasets were networks observed over four time steps. For each dataset we simulated missing links and attributes by removing 20% of the actors completely at random (MAR). We applied each of the five link imputation techniques on all eleven datasets and calculated the AUC of link imputation accuracy for each technique. In general, the imputation accuracy had decreased for all techniques as the number of actors increased (see Figure 5). However, in all experiments ITERGM was much more accurate than any of the alternative techniques.

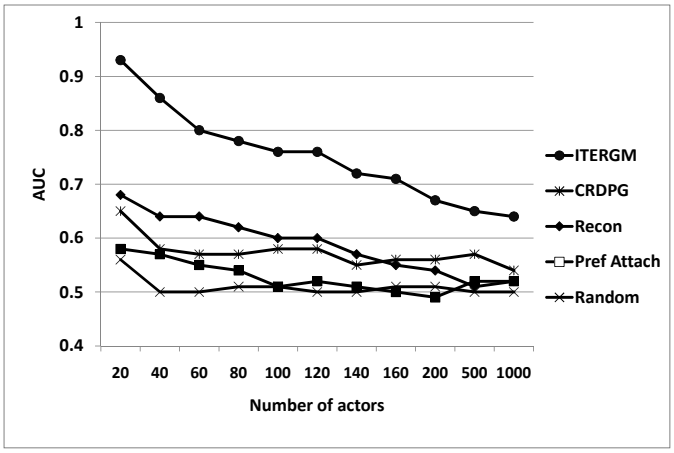

Figure 5: Comparison of the accuracy of link imputation techniques measured in AUC vs. the number of actors on eleven synthetic datasets of increased size. All datasets consist of 4 time steps and the missing data is modeled by randomly removing 20% of actors at each time step.

3.4.2 Real Life Datasets

We conducted an exhaustive set of experiments on two real life datasets *Delinquency* and *Teenagers* (Chapter). On the *Delinquency* dataset ITERGM had achieved convergence on average in three iterations, and four iterations on *Teenagers*. We present the results of the experiments on both real life datasets in Tables 12 and 13. In both real life datasets the ITERGM performed well on the links and attributes' imputation as compared to the baselines. We observed many overlaps in the confidence intervals of the attribute imputation accuracies in the *Delinquency* dataset. However, in many of these instances the confidence intervals of the baseline techniques are rather large whereas the ITERGM is more precise (for example see the experiment of MNAR-score, 60% missing actors). Results on *Teenagers* were notably better, with less overlaps of the confidence intervals. The CRDPG and "Reconstruction" link imputation algorithms also had good results and in almost all cases were the second best choices. Just as we expected, the "Random" algorithm performed poorly (AUC values are close to 0.5).

The confidence interval of "Average" technique was not computed, because its result is deterministic for a given dataset.

Table 12: Links and attributes imputation on the *Delinquency* dataset (simulating five types of missing mechanisms for 20%-60% of missing actors): 90% confidence interval of the AUC and MSE. **Bold** denotes the best result, the underline represents the overlap of the confidence interval with the best result.

Type	%	Link Imputation AUC					
		Random	PrAtt.	Recon.	CRDPG	Mult.Lat.F.	ITERGM
random	20	0.52±0.01	0.60±0.02	0.70±0.00	0.72±0.00	0.74±0.00	**0.78±0.01**
	40	0.52±0.00	0.66±0.01	0.71±0.01	0.70±0.01	0.72±0.00	**0.74±0.01**
	60	0.53±0.01	0.67±0.00	0.65±0.01	0.69±0.01	0.69±0.00	**0.71±0.01**
absent	20	0.50±0.01	0.62±0.01	0.72±0.00	0.72±0.00	0.70±0.00	**0.73±0.01**
	40	0.50±0.02	0.59±0.02	0.65±0.03	0.67±0.01	0.68±0.00	**0.68±0.01**
	60	0.49±0.01	0.57±0.01	0.62±0.01	0.63±0.02	0.64±0.00	**0.65±0.01**
score	20	0.53±0.00	0.67±0.00	0.72±0.01	0.79±0.02	0.78±0.00	**0.80±0.02**
	40	0.52±0.01	0.60±0.01	0.67±0.00	0.66±0.01	0.69±0.00	**0.72±0.01**
	60	0.49±0.00	0.61±0.01	0.68±0.00	0.64±0.02	0.69±0.00	**0.70±0.00**
indegree	20	0.51±0.02	0.54±0.01	0.74±0.03	0.74±0.03	0.77±0.00	**0.79±0.01**
	40	0.54±0.01	0.61±0.00	0.64±0.00	0.74±0.00	0.75±0.00	**0.80±0.01**
	60	0.52±0.00	0.62±0.00	0.65±0.00	**0.72±0.00**	0.66±0.00	0.68±0.00
outdegree	20	0.50±0.04	0.67±0.04	0.86±0.01	0.82±0.06	0.87±0.00	**0.89±0.03**
	40	0.52±0.01	0.61±0.01	0.72±0.02	0.70±0.00	0.78±0.00	**0.81±0.01**
	60	0.51±0.00	0.52±0.01	0.61±0.00	0.65±0.01	0.72±0.00	**0.73±0.01**

Type	%	Attributes Imputation MSE		
		Average	DynMM	ITERGM
random	20	0.19±0.00	0.15±0.00	**0.12±0.00**
	40	**0.25±0.00**	0.46±0.09	0.40±0.15
	60	0.64±0.00	0.60±0.05	**0.51±0.06**
absent	20	**0.24±0.03**	0.31±0.06	0.27±0.06
	40	0.38±0.02	0.48±0.05	**0.35±0.07**
	60	0.78±0.05	0.74±0.03	**0.65±0.06**
score	20	0.49±0.00	0.35±0.25	**0.27±0.02**
	40	0.95±0.00	0.84±0.45	**0.77±0.17**
	60	1.16±0.00	1.01±0.54	**0.77±0.10**
indegree	20	0.27±0.00	0.15±0.01	**0.11±0.00**
	40	0.48±0.00	0.38±0.03	**0.25±0.03**
	60	0.63±0.00	0.62±0.04	**0.54±0.06**
outdegree	20	0.56±0.00	0.48±0.15	**0.43±0.06**
	40	0.58±0.00	0.50±0.08	**0.41±0.08**
	60	0.72±0.00	0.82±0.14	**0.66±0.09**

Table 13: Links and attributes imputation on the *Teenagers* dataset (simulating five types of missing mechanisms for 20%-60% of missing actors): 90% confidence interval of AUC and MSE. **Bold** denotes the best result, the underline represents the overlap of the confidence interval with the best result.

Type	%	Link Imputation AUC					
		Random	PrAtt.	Recon.	CRDPG	Mult.Lat.F.	ITERGM
	20	0.50±0.00	0.53±0.00	0.85±0.00	0.78±0.00	0.83±0.00	**0.86±0.00**
random	40	0.48±0.00	0.50±0.00	0.71±0.00	0.61±0.00	0.65±0.00	**0.71±0.00**
	60	0.51±0.01	0.53±0.00	0.79±0.00	0.74±0.00	0.77±0.00	**0.79±0.00**
	20	0.51±0.00	0.54±0.00	0.78±0.01	0.79±0.01	0.80±0.00	**0.82±0.02**
absent	40	0.49±0.*00	0.52±0.00	0.68±0.00	0.67±0.00	0.69±0.00	**0.75±0.01**
	60	0.49±0.01	0.52±0.00	0.66±0.00	0.64±0.00	0.68±0.00	**0.71±0.01**
	20	0.57±0.00	0.51±0.00	0.79±0.03	0.72±0.00	0.80±0.00	**0.80±0.00**
score	40	0.50±0.00	0.53±0.00	0.75±0.00	0.75±0.02	0.74±0.00	**0.76±0.00**
	60	0.50±0.00	0.51±0.00	0.67±0.00	0.69±0.00	0.72±0.00	**0.73±0.00**
	20	0.48±0.01	0.54±0.00	0.68±0.08	0.69±0.04	0.73±0.00	**0.74±0.00**
indegree	40	0.53±0.00	0.56±0.00	0.69±0.01	0.73±0.00	0.70±0.00	**0.75±0.00**
	60	0.49±0.00	0.53±0.00	0.69±0.00	0.67±0.01	0.69±0.00	**0.70±0.00**
	20	0.50±0.00	0.50±0.00	0.85±0.01	**0.88±0.01**	0.80±0.00	0.82±0.01
outdegree	40	0.49±0.00	0.49±0.00	**0.85±0.01**	0.76±0.00	0.74±0.00	0.77±0.01
	60	0.49±0.00	0.51±0.00	**0.75±0.01**	0.70±0.00	0.71±0.00	0.71±0.00

Type	%	Attributes Imputation MSE		
		Average	DynMM	ITERGM
	20	0.25±0.00	0.11±0.00	**0.10±0.00**
random	40	0.52±0.00	0.52±0.01	**0.51±0.00**
	60	0.84±0.00	0.77±0.01	**0.75±0.01**
	20	0.30±0.01	0.25±0.03	**0.23±0.00**
absent	40	0.92±0.02	0.89±0.02	**0.79±0.02**
	60	1.08±0.01	1.10±0.01	**1.05±0.00**
	20	0.41±0.00	**0.36±0.00**	0.36±0.02
score	40	0.62±0.00	0.60±0.01	**0.54±0.01**
	60	1.38±0.00	1.40±0.03	**1.18±0.04**
	20	0.26±0.00	0.15±0.00	**0.13±0.00**
indegree	40	0.53±0.00	0.36±0.00	**0.33±0.01**
	60	0.75±0.00	0.65±0.04	**0.57±0.02**
	20	**0.16±0.00**	0.17±0.00	0.17±0.00
outdegree	40	0.43±0.00	0.47±0.00	**0.41±0.01**
	60	0.70±0.00	0.74±0.00	**0.53±0.01**

The objective of this chapter is to introduce and evaluate a new log-linear imputation technique for social network surveys and evaluate its accuracy. Here we are mostly concerned with how various imputation techniques compare in terms of precision. However, it is also interesting to see how most accurate imputation methods reviewed in Chapters and affect network statistics. To this end we conducted an additional experiment on the real life dataset *Teenagers*. We investigated how imputations affect outdegree

$$\text{outdegree}(i) = \sum_j N_{ij} \tag{40}$$

and reciprocity

$$\text{reciprocity}(i) = \sum_j N_{ij}N_{ji} \tag{41}$$

of the nodes. Outdegree indicates a student's level of social activity. We expect socially active students to have large outdegrees. Reciprocity measures how a student's friendships are reciprocated, with high reciprocity indicating that friendship feelings between a student and her peers are mostly mutual. In our experiment we removed 20 students from the *Teenagers* dataset using a MAR "absent" mechanism. We imputed missing outgoing links using ITERGM, CRDPG and Multiplicative Latent Factors imputation techniques by running each method twenty times and then taking the median of the outdegree and reciprocity statistics for each student at the third wave panel of the dataset. Boxplots shown in Figure 6 for outdegree and Figure 7 for reciprocity represent the results of this experiment. The first boxplot in both figures is the true statistics distribution of the students modeled as non-respondents. The following boxplots, two through four of each graph, are corresponding outdegree and reciprocity statistics distributions recovered by three imputation techniques. We note the all three technique did a great job recovering distribution of outdegree statistics, as they all managed to correctly predict the median of the distribution and both ITERGM and Multiplicative Latent Factors were fairly accurate in guessing the distribution range.

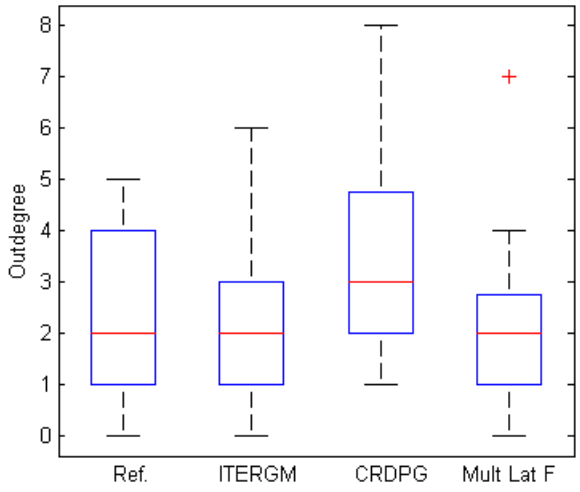

Figure 6: Boxplots of 20 non-respondent students outdegrees from the third time step of the real life dataset *Teenagers*. Non-respondents were modeled as missing at random. First boxplot (Ref) is ground truth of observed outdegrees. Last three boxplots are outdegrees recovered by our technique ITERGM and two other baseline models (CRDPG and Multiplicative Latent Factors model).

From this we draw the conclusion that our imputation technique ITERGM can accurately guess the social activity level of the missing students. Figure 7 shows that recovery of reciprocity is a harder task, as all three imputation techniques overshoot the median estimate of the reciprocity, although ITERGM does that to a lesser degree. However, the overall recovery of missing students reciprocity is good to fair, which is encouraging. This and other experiments described in this chapter suggest that our proposed technique ITERGM can be used in practice by social scientists.

3.5 Scalability

We investigated the runtime of ITERGM based on two sets of experiments. In one experiment we have created a synthetic dataset with 30 actors and 10 time steps. We ran ITERGM to impute this dataset on a increasing number of time steps from 2 to 10 and recorded the time in seconds it took the algorithm to run. In Figure 8 we present the result

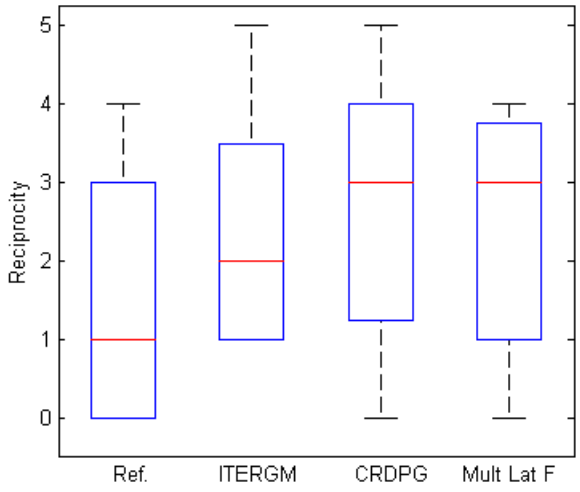

Figure 7: Boxplots of 20 non-respondent students reciprocity statistics from the third time step of the real life dataset *Teenagers*. Non-respondents were modeled as missing at random. First boxplot (Ref) is ground truth of observed reciprocities. Last three boxplots are reciprocities values recovered by our technique ITERGM and two other baseline models (CRDPG and Multiplicative Latent Factors model).

of this experiment. Here, we clearly observe a linear trend of algorithm runtime in terms of number of survey panels.

We conducted a similar experiment on 4 survey panels. This time we held the number of surveys constant but were increasing the number of actors from 30 to 100 in 5 actor increments. We ran the imputation algorithm on the resulting dataset and recorded the time in seconds it took to run. The results of this experiment are shown in Figure 9. In this experiment, we observed the quadratic term of algorithm runtime in terms of number participating actors. The quadratic scalability in terms of number of actors is not surprising because the algorithm has to consider $k^2 - k$ number of relationships (k is the number of actors in the social networks).

The biggest dataset we had conducted experiments on was a synthetic dataset containing 1000 actors, and it is possible to go higher than that. However, it was mentioned in the literature [81] that networks containing more than a few hundred nodes are impractical for real life social surveys. Consider a class which has 500 students. If we ask each stu-

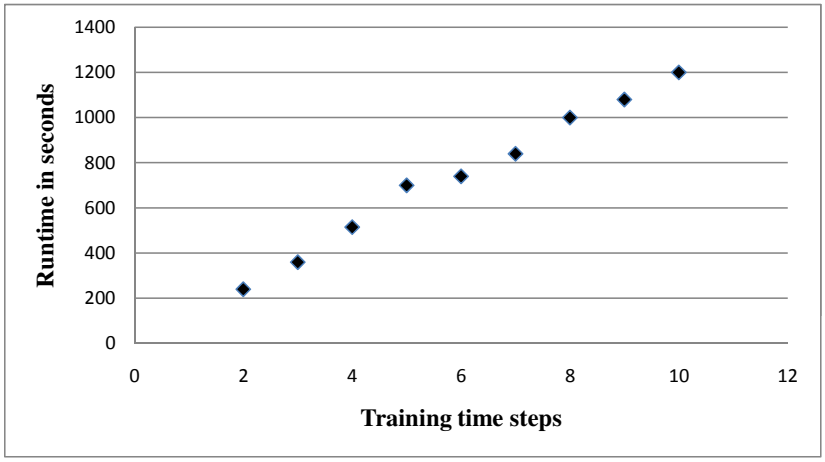

Figure 8: ITERGM runtime in seconds vs. number of surveys.

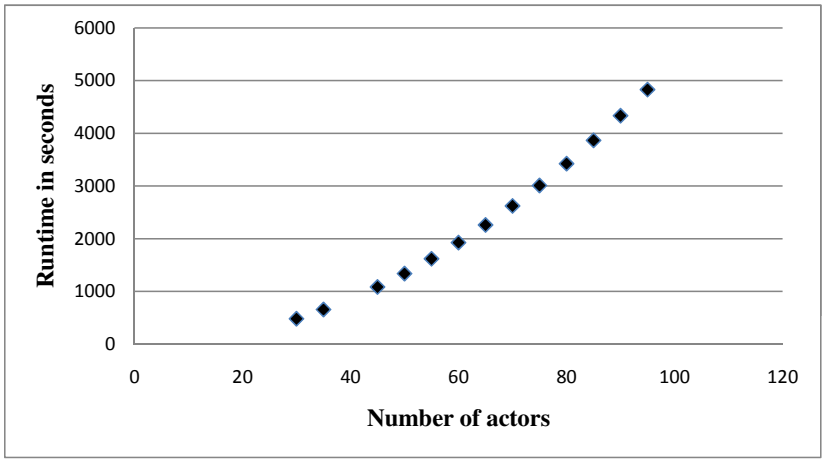

Figure 9: ITERGM runtime in seconds vs. number of actors.

dent of such class to identify all their friends, we know that the density of such a network will be very low. This is because we would expect each student to have at most twenty or thirty friends and many students will have much less friends than that. Human beings are naturally incapable to effectively handle hundreds of close social relationships at the same time. Therefore, the structural 0's in such a network most of the time will indicate that two pupils had never had opportunity to get to know each other rather than a social dislike. Such observation holds true for all large networks. For example, we know for sure that users of Facebook are not aware of the existence of the majority of other users and will never establish links with them.

Recent work [36] on tERGM, a predecessor of etERGM, had suggested the use of a specially factorized statistics which are expressed in the form:

$$\mathbf{\Psi}(N^t, N^{t-1}) = \sum_{ij} \mathbf{\Psi}_{ij}(N^t_{ij}, N^{t-1}) \tag{42}$$

The use of such statistics avoids the expensive Gibbs sampling process. Instead, it is replaced by direct computation of the expectation step of the learning algorithm. etERGM, which is a cornerstone of our approach, introduces statistics which cannot be factorized into the form specified by (42). These statistics model a homophily selection process often found in social networks. Homophily is often expressed as "birds of a feather flock together", meaning the actors with similar attributes are more likely to form a social relationship. Homophily is an essential part of etERGM, while the factorization method in Equation (42) restricts its expression. This makes it implausible to use such factorization in etERGM and also in our approach.

There are techniques, such as fast approximation, which could speed up the learning and inference process of our approach. However, before applying such techniques one should consider the specifics of collecting the social network surveys. The real life datasets in our work and other similar real life datasets have one thing in common. It takes months, sometimes years, to collect the temporal social data. Such long collection time is necessary because the social links between humans are relatively stable and do not change that often. Therefore, the prevalent time step granularity of such surveys ranges from months to years. Given the time and cost it takes to conduct the surveys, the imputation accuracy has much greater importance than speed of imputation, as long as the imputation algorithm takes a reasonable time to run. Our proposed approach runs in the order of minutes for the most common temporal social surveys, and it is more than adequate. Hence, we have avoided fast approximation lest the accuracy of the imputation suffer.

3.6 Discussion

The proposed log linear approach ITERGM compared favorably in empirical evaluations against alternative techniques. The disadvantage of this approach is that it cannot infer the first time step. We investigated the possibility of training an ITERGM model on the reversed time sequence of the surveys, so that inference of the first time step would be possible. However, we have encountered the degeneracy of the link prediction model not uncommon in exponential random graphs [72].

CHAPTER 4

MORTALITY PREDICTION IN SOCIALLY LINKED POPULATION

In the previous chapters we discussed how log linear models can be used for prediction and imputation in temporal social networks. However, the application of log linear models is by no means is limited to dynamic networks. Here, we demonstrate how novel log linear technique can be used to improve mortality prediction of the socially linked group of people represented by a static graph. To be more specific, we examine the population of hemodialysis patients who form social connections in the places where they received treatment.

Chronic kidney disease (CKD) is a significant public health problem that affects more than 30 million adults in the United States, 354,600 of whom are receiving hemodialysis for end stage renal disease (ESRD) [65]. With a staggering 60% 5-year mortality rate, hemodialysis patients have one of the worst outcomes for any chronic disease including HIV/AIDS and breast cancer [65]. African Americans suffer a disproportionate burden of CKD and end stage renal disease with adjusted incidence and prevalence rates 4-6 times higher than whites. This burden is further exacerbated by racial disparities that render African Americans less likely to be referred for transplantation, the treatment of choice, less likely to complete pre-transplantation evaluation, and less likely to be transplanted [5, 6, 28, 45, 78, 89, 95].

If a kidney transplant is not immediately available, the majority of patients with ESRD undergo hemodialysis three times a week at a freestanding clinic which is typically a large room with 20-30 patients undergoing treatment simultaneously. The existence and influence of social networks within hemodialysis clinics was first suggested in 2002 by Klassen and colleagues who observed that some patients viewed racial relations as more positive in

the dialysis units than in other settings [52]. Yet, the influence of social networks in kidney disease populations remains vastly understudied [16, 34].

Phosphorous level is a known indicator of adherence to the strict dietary regiment vital for the survival of hemodialysis patients [97]. Unlike some other biomarkers, phosphorous level in the bloodstream is not confounded by other factors and is dependent only on dietary intake and medication adherence [80]. There are at least two ways that a social network may affect the serum phosphorous level of patients receiving hemodialysis on the same schedule at the same facility. First, patients talk about their disease, how they cope with it, what they eat and what they think about their dietary restrictions. This information sharing is a form of social interaction that could result in correlated measurements such as contemporaneous phosphorous spikes. Second, although it is prohibited, some patients bring in snacks and share them with their friends during treatment. Sharing of snack foods which tend to be high in phosphorous could also explain synchronous spikes of phosphorous in two patients.

We posit that the informational gain obtained from adding social interactions to the prediction model will improve the accuracy of mortality prediction. We use phosphorous correlations to demonstrate the existence of the social links in a predominantly African American hemodialysis population. We also show that this information improves mortality prediction by novel log linear model.

4.1 Data and Data Preparation

This study examines 116 hemodialysis patients from two outpatient clinics in Philadelphia, PA, from the beginning of May 2008 to the end of January 2011. The dataset is comprised of repeated biomarker measures extracted for the 116 subjects in this study from the medical database and aggregated over 11 calendar quarters together with survey data collected from participating patients (response rate 91%) between August 2008 and January 2009 using the Temple University Dialysis Patient Transplant Questionnaire (DPTQ) [34].

Blood was collected and tested every 3-4 weeks and additional tests were done off the

Table 14: Population census broken down by shift and location, the number in parenthesis is how many patients died within a group.

Shift	Days	Clinic#1	Clinic#2
AM Shift	Mondays Wednesdays Fridays	17(2)	11(2)
	Tuesdays Thursdays Saturdays	20(7)	21(3)
PM Shift	Mondays Wednesdays Fridays	12(2)	10(2)
	Tuesdays Thursdays Saturdays	10(4)	15(1)

regular schedule (e.g., during hospitalization) as prescribed by attending physicians. This resulted in approximately 66 specimens per patient with some variation due to a small number of patients with missing data early in the study period, 23 deaths, and a number of hospitalizations. Each blood specimen yielded approximately 60 test results (biomarkers). As detailed in our study [67], we relied on domain expert opinion and logistic regression-based feature selection to reduce the 60 biomarkers to four mortality predictors frequently cited in the hemodialysis literature. These are albumin, hemoglobin, potassium and sodium [91].

The same feature selection technique was used to select the following three mortality predictors from the survey data: patient age at interview, self-reported health, and perceived racial bias. The choice of selected questionnaire variables was only based on their strong association with patients' mortality. Descriptive statistics for the survey and medical data for the 116 study participants are reported in [34].

Within each clinic, participating patients were dialyzed on four alternative schedules: mornings or afternoons on Mondays, Wednesdays and Fridays, and mornings or afternoons on Tuesdays, Thursdays and Saturdays. No patients switched clinics or schedules during the study period. Thus, eight distinct patient groups (four per clinic) were created by cross-classifying the shifts (morning and afternoon) with the two three-day treatment options (Monday, Wednesday, and Friday vs. Tuesday, Thursday, and Saturday). Also, any interactions between patients occur within a single group. Table 14 shows the population and mortality data disaggregated by treatment days, shift and clinic. According to Fisher's

exact test [29] there is no statistically discernible difference between groups' death rates, with p-value = 0.35 .

4.2 Hemodialysis Social Networks

The social interaction between patients during dialysis suggests the presence of social networks within clinics. To construct links within each of the eight patient groups, we computed a Pearson correlation coefficient (a well known and widely adopted tool for establishing linear dependency between variables and mapping networks in various domains [86], [58]) for all possible pairwise relationships within a single group based on contemporaneous phosphorous measurements. A social link was established between two patients if both had at least ten contemporaneous blood tests and a statistically significant linear dependency at $p < 0.05$. Figure 10 illustrates a strong correlated relationship with many simultaneous ups and downs in the phosphorous time series for two patients linked by our technique.

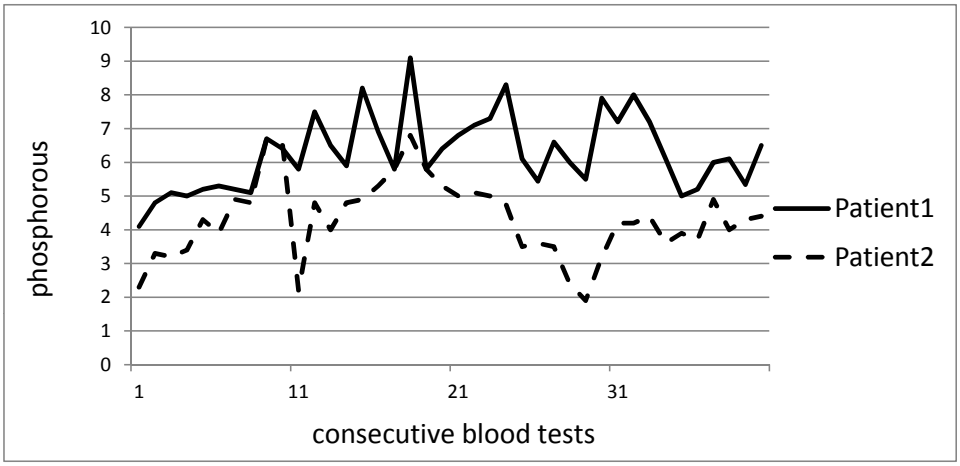

Figure 10: Example of correlated phosphorous time series between two connected patients, p-value = 0.004

A statistically significant correlation between two variables does not imply their dependence. It can be so that correlation is spurious and had occurred by chance. We verified

Table 15: Number of links, density, randomized test t-value and confidence level of each discovered network based on 2,000 randomizations. MWF denotes Mondays, Wednesdays and Fridays schedule and TThS denotes Tuesdays, Thursdays and Saturdays.

Clinic	Shift	Days	# of links	Density	t-value	Confidence
Clinic#1	AM	MWF	10	0.07	1.221	73.0%
		TThS	14	0.07	1.879	94.0%
	PM	MWF	8	0.12	2.647	99.2%
		TThS	7	0.16	2.637	99.2%
Clinic#2	AM	MWF	10	0.18	4.667	99.9%
		TThS	18	0.09	2.385	98.5%
	PM	MWF	11	0.24	5.975	99.9%
		TThS	14	0.13	3.770	99.9%

that discovered links were not spurious with two experiments, the first of which uses a Randomization Test [27]. This test is a powerful non-parametric statistical technique designed to accept or reject a null hypothesis $H_0 : m_1 - m_2 = 0$ by sampling from multiple randomized populations derived from the original data, and using the multiple measurements sampled from these randomized sets to calculate a test statistic (t test in this study) to ascertain the validity of a hypothesis. Here, the number of discovered links within a single group m_1, is equal to the number of links in a random population m_2. In other words, the number of discovered links is random. We tested this hypothesis for every network by randomly permuting all the phosphorous measurements within each of the eight patient groups 2,000 times. With each permutation, we built a social network according to the earlier described criteria and counted links within the resulting network. The t-statistics based on the 2,000 samples drawn from the randomized networks with corresponding confidence levels, and the network censuses and link densities are presented in Table 15. Six of the eight confidence levels are statistically significant at $p < 0.05$ suggesting that the constructed networks are not random. We therefore reject the null hypothesis.

In the second experiment we attempted to discover social links between the patients who were not on the same schedule. We separated the dataset into two partitions. The first

partition consisted of patients visiting clinics on Mondays, Wednesdays, Fridays (MWF) and the second partition consisted of patients attending on Tuesdays, Thursdays and Saturdays (TThS). Such partitioning was necessary because blood samples taken from patients on the MWF schedule were not contemporaneous with samples taken from groups on the TThS schedule. We then applied our link discovery technique on each partition and counted links connecting its four groups. In the first partition there were 125 cross-group relationships out of a possible $2,003$, which gives a network density of 0.06. In the second partition there were 236 cross-group links out of a possible $3,815$, resulting in the similar density of 0.06. It is remarkable that density of both cross-group networks within each partition is virtually the same and it is less than the intra-group densities reported in the Table 15. We hypothesize that network density of 0.06 is heuristic; indicative of purely random and coincidental correlations of patients' phosphorous level. This suggests that simultaneous movements of patients' phosphorous level can only be explained by processes within each group reported in the Table 15. This fact minimizes possibility that a certain phosphorous movement pattern could have conferred a mortality risk in unrelated patients.

To further validate the hemodialysis social networks , an additional experiment is conducted on data from the Monday, Wednesday, Friday (MWF) morning shift at Clinic#1 from which five patients with unrelated phosphorous were selected. In a similar fashion five patients were selected from the same clinic's MWF afternoon shift and five additional patients from the MWF afternoon shift at Clinic#2. The three groups were specifically selected from the MWF schedule to ensure that the phosphorous measurements were taken at the same time. A random social network was constructed from the resulting population of 15 patients.

By construction, there are only 25 possible links between groups 1 and 2, 2 and 3, and 1 and 3. Out of total of 75 possible relationships, we identified four links (right panel of Figure 11) yielding a network density of 4/75 = 0.053. The Randomization Test confirmed that the average number of links in the random network illustrated in the right panel of Fig-

ure 11 is not statistically different from four (t-value is -0.45 based on 2,000 permutations, H_0 is not rejected), providing further evidence of the validity of phosphorous based links.

Figure 11: Comparison of two real life networks with a random network. Left and middle panels show real life networks of patients with correlated phosphorous in Clinic#1. The right panel was obtained by randomly connecting patients with uncorrelated phosphorus to each other from three MWF hemodialysis groups. The random network is less dense than actual networks. White nodes denote deceased patients.

The left and middle panels of Figure 11 show the networks of patients with correlated phosphorus for MWF morning and afternoon shifts in Clinic#1. The white nodes represent patients who have died. The right panel in Figure 11 was obtained by randomly connecting three groups of five patients who have unrelated phosphorous. The randomly constructed network appears less dense than real networks depicted in the left and middle panels.

4.3 Log Linear Mortality Prediction Model

A baseline mortality prediction was obtained by a logistic regression model defined as

$$\log \text{odds}(p_{qr} = 1) = \boldsymbol{\beta}_1 \boldsymbol{u}_{q,r-1} + \boldsymbol{\beta}_2 \boldsymbol{u}_{q,r-2} + \boldsymbol{\beta}_3 \boldsymbol{v}_q \tag{43}$$

Here, p_{qr} denotes the log odds of patient q's death during calendar quarter r. The dependent variables $\mathbf{u}_{q,r-1}$ and $\mathbf{u}_{q,r-2}$ are the patient's vectors of biomarker averages for the preceding non-overlapping (adjoined) calendar quarters $r-1$ and $r-2$. The variable $\boldsymbol{v}_q$ is a vector of the patient's covariates from medical record and survey data. Finally, variables $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ and $\boldsymbol{\beta}_3$ are vectors of logistic regression coefficients.

The baseline mortality prediction model described by Equation (43) assumes that death observations are independent. It does not take into account the social mechanisms which might influence a patient's prognosis. This is why we propose a new log linear technique which, contrary to the baseline model, takes advantage of the discovered social interactions. Our mortality prediction model describes a patient i by a vector $\boldsymbol{m}_i^{1\times d}$ of latent coordinates in a d-dimensional social space. We do not assert the meaning of what coordinates represent and we simply assume that a weighted vector dot product of latent coordinates of two patients yields a "social" value, denoted as $\hat{e}$. We hypothesize that this value is one of the factors which can influence a patient's well-being through the mechanisms discussed earlier. The real-valued weights used in latent coordinates multiplications are also latent. The value of a social influence $\hat{e}$ experienced by a patient is modeled as the sum of the weighted dot product multiplications of a patient's latent coordinates and the coordinates of that patient's neighbors in a social network. More formally, we define a social influence value $\hat{e}$ for a patient q as

$$\hat{e}_q = \sum_{s \in V(q)} W_{qs}\, \boldsymbol{m}_q \cdot \boldsymbol{m}_s \tag{44}$$

where $\boldsymbol{W} \in R^{n\times n}$ is a symmetric sparse weight matrix, n is number of patients, and $\boldsymbol{m}_i \in R^d$, $i \in 1 \cdots n$ is a set of latent coordinates. Finally, V is a set of patients directly linked to a patient q by a social network. The absolute value of the weight models the strength of the positive or negative influence one person might have on another. Another implication of this model is that a patient not connected to anyone else will have a social factor value 0. We augment the baseline prediction model described by Equation (43) by incorporating Equation (44). Thus, our proposed model, named Social Latent Vectors (SLV), is defined as:

$$\log \operatorname{odds}(p_{qr} = 1) = \boldsymbol{\beta}_1 \boldsymbol{u}_{q,r-1} + \boldsymbol{\beta}_2 \boldsymbol{u}_{q,r-2} + \boldsymbol{\beta}_3 \boldsymbol{v}_q + \boldsymbol{\beta}_4 \hat{e}_q \tag{45}$$

where $\hat{e}_q$ is described by Equation (44). The latent weights $\boldsymbol{W}$ and latent coordinate vectors

$\boldsymbol{m}$ of our model are both required for computation of a patient's social factor $\hat{e}$. We learn $\boldsymbol{W}$ and $\boldsymbol{m}$ as well as regression coefficients $\boldsymbol{\beta}$ via the iterative Expectation-Maximization Markov Chain Monte Carlo (EM MCMC) procedure described in the following chapters.

4.3.1 Algorithm Characterization

In Algorithm 2, we iteratively sample link weights and patient coordinates in the latent space. Whenever a sample is accepted, we recalculate the corresponding social influence values $\hat{e}$. When the MCMC chain ends, we average accepted samples and use the averaged values to update the social factor values $\hat{\boldsymbol{e}}$ (Equation 44). Finally, the model regression parameters $\boldsymbol{\beta}$ are updated through convex optimization (Equation 45). The algorithm repeats until convergence criteria are met. One of the advantages of our approach is that evaluation of a single sample does not require recalculation of the entire dataset's likelihood. For example, if we sample a latent coordinate vector $\underline{\boldsymbol{m}}_{q,(t)}$ of a patient q at iteration t, we only have to evaluate the density described by Equation (46):

$$\prod_{s\in\{q,V(q)\}}\prod_{r\in Q(s)} P(p_{sr}|\underline{\boldsymbol{m}}_{q,(t)}, \boldsymbol{m}_{-q,(t)}, \boldsymbol{\beta}_{(t-1)}, \boldsymbol{W}_{(t)}, \boldsymbol{D}_{sr}) \tag{46}$$

In Equation(46), $V(q)$ denotes the set of patients directly linked by a social network to the patient q. Therefore, the set $\{q, V(q)\}$ is a patient q plus all of the patient's social network neighbors. $Q(s)$ is a set of calendar quarters when patient s tests were available and p_{sr} is an event of death or survival during calendar quarter r. We define $\boldsymbol{D}_{sr}$ as the set of predictive variables for patient s at the calendar quarter r: $\{\boldsymbol{u}_{s,r-1}, \boldsymbol{u}_{s,r-2}, \boldsymbol{v}_s\}$. Here, $\boldsymbol{u}_{s,r-1}$ are four biomarkers' averages for a calendar quarter prior to quarter r, $\boldsymbol{u}_{s,r-2}$ are biomarker averages two quarters ago and $\boldsymbol{v}_s$ are patient predictive variables from the survey. $\boldsymbol{m}_{-q,(t)}$ is a set of all patients' latent coordinate vectors minus vector coordinate $\underline{\boldsymbol{m}}_{q,(t)}$ of patient q which we sample. Finally, $\boldsymbol{W}_{(t)}$ is the symmetric matrix of latent weights at iteration t. We restrict evaluation of the density only to a sampled patient and patients immediately

connected to the sampled patient by the network because in our model all other patients are conditionally independent given the rest of the graph. Also, when sampling $\underline{\boldsymbol{m}}_{q,(t)}$ we only have to consider patients who have at least one social link because the social factor value of an isolated patient is always 0. The same applies to the sampling of weight matrix $\boldsymbol{W}$. We only sample matrix entries which correspond to an existing link.

4.3.2 Algorithm Regularization

Our model is flexible because of its latent parameters $\mathbf{W}$ and $\mathbf{m}$. Potential overfit problems due to so many parameters can be addressed by the l_1 regularization technique for MCMC sampling [30]. This technique was applied previously as the regularization path for the model space of Bayesian Model Averaging (BMA) but it can be easily adapted for a single model. In our work we define a regulatory parameter α which sets the acceptable range $[-\alpha, \alpha]$ for a sampled latent parameter. During MCMC sampling we reject a sample if its value falls outside the regularized range. The α parameter has to be calibrated in order to avoid overfit. If the range is too small the model will not reach local maxima. In our implementation, α was calibrated such that during multiple restarts the chance of observing overfit on the training data was 20%. Whenever we observed overfit, the MCMC chain was restarted.

4.3.3 Algorithm Convergence

The convergence criteria for Algorithm 2 is sufficiently small error, defined as:

$$\sum_{i=1}^{n} abs(\hat{e}_{i,(t)} - \hat{e}_{i,(t-1)}) \tag{47}$$

where n is the number of patients. During each iteration of Algorithm 2 marked by the outer loop variable t we measure the change in social factor values, compared to the previous iteration. If the sum of absolute differences of all factors is small, it means that the social

Algorithm 2 Social Latent Vectors

Input: $\mathbf{u}$ - biomarkers time series
$\mathbf{v}$ - survey questionnaire's answers
n - number of patients
$N^{n\times n}$ - social network, where $N(i,j)=1, N(j,i)=1$ denotes a link between patients
i and j and $N(i,j)=0, N(j,i)=0$ denotes link absence
$\mathbf{p}$ - death/survival events
d - dimensionality of latent coordinate space
C - number of MCMC samples

Output: $\hat{\mathbf{e}}$ - patients social factors
$\boldsymbol{\beta}$ - logistic regression coefficients

- initialize iteration counter: $t=1$
- randomly initialize: latent weight matrix $W^{n\times n}_{kl,(t)}$ for $k,l \in 1\dots n \iff N(k,l)=1$
- randomly initialize: latent patient coordinates $\mathbf{m}^{1\times d}_{k,(t)},\ k \in 1\dots n$
- initialize social factors: for $k \in 1\dots n$ calculate $\hat{e}_{k,(t)}$ from $\{\mathbf{W}_{(t)}, \mathbf{m}_{(t)}, N\}$ (Equation 44)
- initialize coefficients $\boldsymbol{\beta}_{(t-1)}$ by running logistic regression on $\{\mathbf{u}, \mathbf{v}, \mathbf{p}, \hat{\mathbf{e}}_{(t)}\}$ (Equation 45)

for t in $1\dots$ until convergence **do**
  for for c in $1\dots C$ **do**
    for i in $1\dots n$ **do**
      sample latent coordinates: $\mathbf{m}^{c}_{i,(t)} \sim P(\underline{\mathbf{m}}_{i,(t)}|\mathbf{m}_{-i,(t)}, \mathbf{W}_{(t)}, \mathbf{p}, \mathbf{u}, \hat{\mathbf{e}}_{(t)}, \boldsymbol{\beta}_{(t-1)}, N)$
    end for
    for i,j in $1\dots n$ if $N(i,j)=1$ **do**
      sample latent weights: $W^{t,c}_{ij} \sim P(\underline{W}_{ij,(t)}|\mathbf{W}_{-ij,(t)}, \mathbf{m}_{(t)}, \mathbf{p}, \mathbf{u}, \hat{\mathbf{e}}_{(t)},\ \boldsymbol{\beta}_{(t-1)}, N)$
    end for
  end for
  average sampled coordinates: for $k \in 1\dots n$ calculate $\hat{\mathbf{m}}_{k,(t)} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{m}^{c}_{k,(t)}$
  average sampled weights: for $k,l \in 1\dots n \iff N(k,l)=1$, $\hat{W}_{kl,(t)} = \frac{1}{C}\sum_{c=1}^{C} \hat{W}^{c}_{kl,(t)}$
  update social factors: for $k \in 1\dots n$ calculate $\hat{e}_{k,(t)}$ from $\{\hat{\mathbf{W}}_{(t)}, \hat{\mathbf{m}}_{(t)}, N\}$ (Equation 44)
  obtain coefficients $\boldsymbol{\beta}_{(t)}$ by running logistic regression on $\{\mathbf{u}, \mathbf{v}, \mathbf{p}, \hat{\mathbf{e}}_{(t)}\}$ (Equation 45)
end for

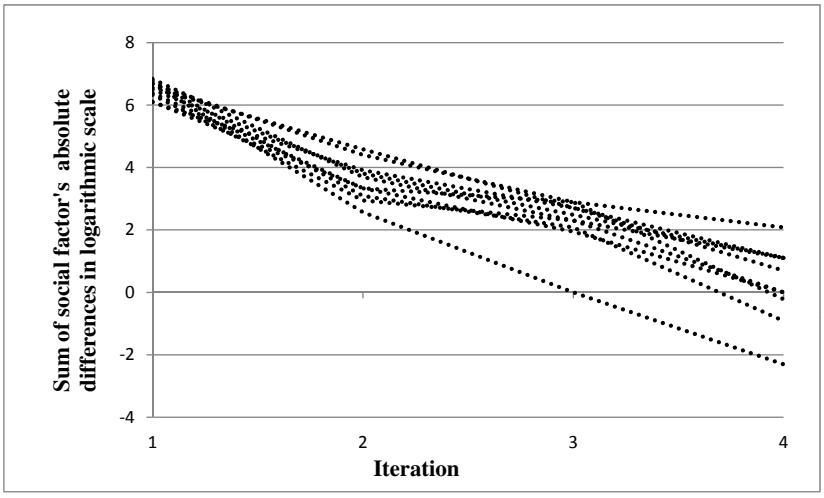

Figure 12: Convergence of Algorithm 2 based on 10 runs. X-axis denotes the iteration (a variable t of Algorithm 2). Y-axis is the logarithmic scale divergence : $ln \sum_{i=1}^{n} abs(\hat{e}_{i,(t)} - \hat{e}_{i,(t-1)})$

factors have stabilized and we can stop the training process. In Figure 12, the convergence characteristics of our algorithm are shown on a logarithmic scale. Here, each time series line denotes a single training run of the algorithm, the X-axis is the iteration number t and the Y-axis is the natural logarithm of the convergence check value expressed by Equation (47). As seen in Figure 12, our model exhibits good convergence properties with very small changes in patients' social factors after the third iteration. After the initial two iterations, a social factor for a patient had changed on average by $\frac{e^2}{116} = 0.06$. This change is negligible compared to the change in the first iteration which was around $\frac{e^6}{116} = 3.47$ (116 is the number of patients).

4.3.4 Algorithm Scalability

To evaluate the scalability of our algorithm, we measured the algorithm's runtime in terms of length of historical records available for patients and population size. In those experiments we held the length of the MCMC chain steady at 30,000 draws (value C in Algorithm 2), 20,000 of which were throw-away burn-in samples. Also, the parameter controlling the dimensionality of the latent space was set to $d = 3$ to keep the model parsimonious. We first

studied the effect of the population size on the algorithm's performance. This is achieved by starting from the complete dataset and gradually deleting patients and their associated records. After every round of deletion we ran our algorithm and recorded the time it took to converge. Our dataset is temporal such that every patient has more than one record. The deletion was done so that the average number of records per patient remained constant (at the value found in the complete dataset $\frac{862\ records}{116\ patients} = 7.4$ records per patient) allowing for a controlled experiment. The results of this experiment are presented in Figure 13. The X-axis in Figure 13 is the time the algorithm took to run and the Y-axis is population size. The linear trend implies that our approach can scale to a much larger dataset. By our estimation, it would take the model less than 12 hours on a typical desktop computer to train on a hypothetical dataset of 10,000 patients.

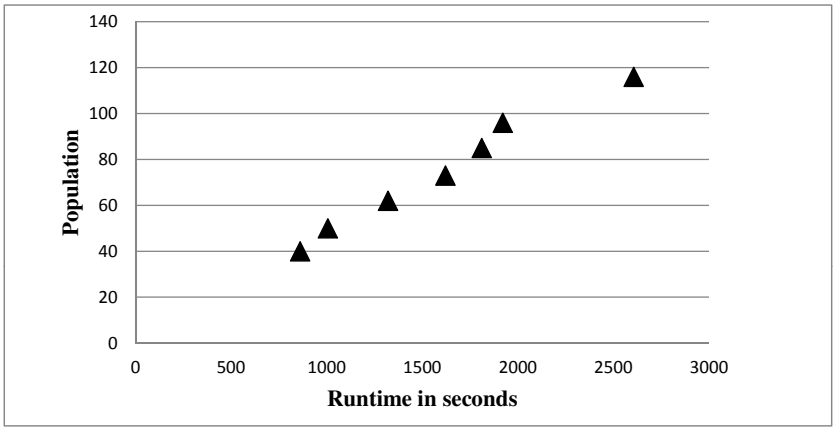

Figure 13: Runtime of proposed algorithm in terms of population size. Runtime is shown in seconds versus population size.

In the second experiment which also suggests a linear trend, we measured the impact of a patient's temporal history on algorithm performance by applying a similar gradual deletion technique. This time we held the patient's population constant while gradually deleting the patient's history. The runtimes of this experiment are presented in Figure 14 where the Y-axis is the average number of temporal records per patient.

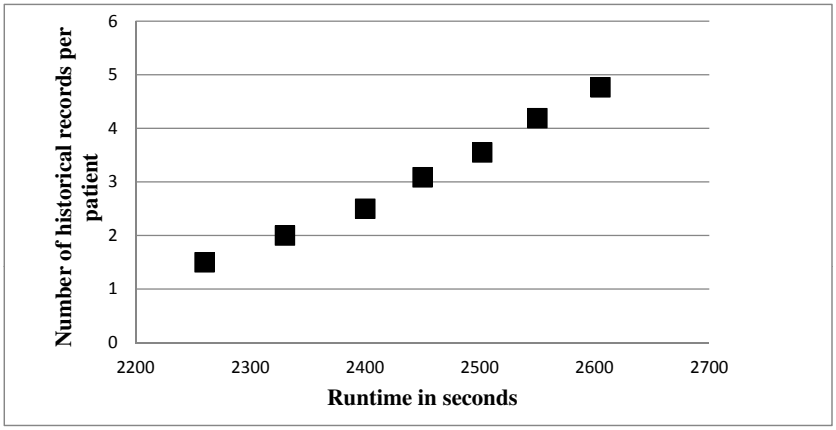

Figure 14: Runtime of proposed algorithm in terms of patient's available medical history. Runtime is shown in seconds versus average length of a patient's history.

4.3.5 Estimation of Social Space Dimensionality

The parameter d represents the dimensionality of our model's social space. In Table 16 we provide empirical estimation of its optimal value. In this experiment we trained the proposed model on 552 records (64% of total) and evaluated its accuracy on 310 records (36%) in the test. The bold value in Table 16 denotes the most accurate prediction against the test records measured by the AUC. The results of this experiment suggest that our model benefits from the parsimonious dimensionality of social space $d < 5$.

Table 16: Mortality predictions of SLV for different dimensions of social interaction space ($d \in 1, 3, 5, 10$). AUC averages and one standard error are reported based on 20 runs of algorithms. Bold value denotes the most accurate prediction for a pre-specified value of parameter d.

d	1	3	5	10
SLV	0.66±0.01	**0.67±0.02**	0.64±0.02	0.60±0.04

4.3.6 Review of Related Work

Prior work in the machine learning domain related to our approach has been concerned with the prediction or imputation of social links. To the best of our knowledge, there is no published work that directly addresses the prediction problem described here. Nevertheless,

there are works by Hoff that also model social actors using latent coordinates [44, 42, 43]. In 2002, Hoff proposed to model the probability of a social link in terms of logistic regression with latent parameters defining the social coordinates of an actor [44]. Hoff's follow-up work [42] expanded the prediction model by introducing higher order dependencies via bilinear effects. In Hoff's latest work [43], a more general model is discussed where link probabilities are represented by multiplying the actors' vectors by a parameter weight matrix, the weight matrix there is different from the weight matrix in our model. In our model, matrix $\boldsymbol{W}$ consists of univariate real valued weights used in vector multiplications. Also, the dimensionality of the weight matrix is the same as that of the latent vectors so that the value of a link's contributing prediction factor is expressed as: $\boldsymbol{z}_i' \boldsymbol{D} \boldsymbol{z}_j$. Similar to our model, Hoff's models [43, 44, 42] consider the actor's covariates, but the problems addressed are inverse to ours. These models predict occurrence of a social link given actors' covariates. In our work, we assume that links are already known and we apply them to predict an important covariate.

In Airoldi et al.[3], a link probability is modeled via latent K-class blockmodel stochastic components, where the probability of a link is dependent on the membership of two actors in an unknown latent class. The learning of latent parameters is similar to ours, and done through sampling of latent parameters followed by convex optimization. However, it investigates the confidence sets for link prediction models, which is a different problem. Yet another work [60], imputes social links using a learning technique vastly different from ours to model actors via latent coordinates.

4.4 Results

Our objective was to test the hypothesis that our mortality prediction model for hemodialysis patients is more accurate than baseline models which do not account for social interactions. Data from the eight patient groups preprocessed as described in the previous chapter were integrated into a single graph which is divided temporally into disjoint training and

Table 17: AUC average and one standard error based on 20 experiments. Bold values denote the best predictor.

SLV	SLV(Random)	Logistic	Logit Boost	RBF	AD Tree
0.67±0.02	0.63±0.01	0.63±0.00	0.58±0.00	0.63±0.00	0.58±0.00

test parts. The test dataset contained 36% (310 records) of each patient's latest historical records and the earliest 64% (552 records) were used for training the prediction models. Such a division strategy ensured that every patient appearing in the test dataset was also present in training. This was necessary because a patient appearing in test but not in training could not be made part of the constructed social network, which can only be derived from historical records. Our approach requires a significant amount of historical data and it can only predict a patient who attended the hemodialysis clinic for an extended period. This observation makes sense because a newcomer patient will need months to develop social ties with other patients in the group.

We compared the accuracy of our algorithm against various baselines. The first baseline SLV(Random) was our model with a randomly permuted underlying graph replacing a network based on phosphorous correlation, it was introduced as an assurance that the social network upon which our model relies is not spurious. The second baseline Logistic Regression was the model presented in Equation 43 and it is synonymous to mortality prediction model presented in [67]. This simple baseline assumes independence between outcomes because it does not consider social links.

The other baseline alternatives were an additive logistic regression classifier Logit Boost [33], a radial basis functions network [69] and alternative decision tree algorithm ADTree [32], none of which accounted for social interactions. For low dimensional latent space of social interactions $d \leq 5$ our model was the most accurate, suggesting that the social networks are valid (Table 17 reports results for $d = 3$). Further evidence of the validity of our social network is seen by comparing the SLV and SLV(Random). In SLV(Random), the underlying social network was random, and this explains why it is less accurate than

SLV, which relied on true networks. The higher accuracy of our model as compared to the baseline models suggests there is an informational gain which can be leveraged by mining hemodialysis social links.

4.5 Discussion

To the best of our knowledge only one published work in the nephrology literature has modeled social networks in a hemodialysis population [16]. The major difference between our work and [16] is that our model is based on discovered links within a hemodialysis clinic whereas in [16] the aggregated statistics of a patient's social network outside of the clinic were considered. We are also the first to use phosphorous correlations to demonstrate the existence of social links in a predominantly African American hemodialysis population. Furthermore, we show that this information improves mortality prediction. Identifying social networks also has the potential for novel interventions to improve adherence not only in hemodialysis clinics but other health care settings. Our study, however, must be considered in the context of its limitations. The results are based on convenience sample of 116 urban dwelling and predominantly African American patients at two hemodialysis clinics. Larger samples from diverse hemodialysis clinics are needed to explore the relevance of our findings in other settings and to hemodialysis patients in general. Also, a validation of discovered social links is warranted. At this point it is too late to do a retrospective survey to cross-validate discovered links against the real patients' sentiments, but we shall address this in the follow-up study. Nevertheless, our findings underscore the value of including patient social network data in hemodialysis patient mortality prediction models.

CHAPTER 5

WEIGHTED TEMPORAL EXPONENTIAL RANDOM GRAPH MODEL FOR TEMPORAL PHENOTYPE DISEASE NETWORKS

The network analysis were used in the past for human phenotypes studies. In this chapter we present new log linear statistical model for temporal Phenotype Disease Network (PDN) dataset derived from monthly aggregations of hospital admissions. The application of this model is different from everything we reviewed insofar the data sets we studied in Chapters , and were human social networks. The proposed model in this chapter will be applied to the disease networks, where each node of the network represents the type of diseases, not a human. Thus the model for such network should demonstrate applicability of the log linear model beyond the realm of social networks. The new technique was applied to predict future prevalence rate PDN describing two hundred fifty major human ailments categories. Our objective is to determine whether statistical model based on a set of informative statistics can be a better fit than trivial approaches in describing PDN temporal characteristics. To this end we evaluated prediction accuracy of our model against latest and commonly known prediction techniques for weighted graphs. The empirical results based on prediction of disease prevalence rates in various geographical regions of the United States suggest that our model is informative of the observed temporal dataset.

We are not the first in our attempt to study PDNs using a network science approach. In [21] it was proposed to use triads census statistics [62] to predict unobserved links in the static bi-modal diseases network where diseases are linked by both rate of co-occurrence and by genetic studies. The empirical results in [21] were encouraging and demonstrated the promise of applying statistical methods to discern patterns in PDN. The main difference between our work and Davis et. al [21] is that we are interested in the temporal aspect of

PDN, whereas the graph constructed in [21] was static. The work by [92] noted that PDN as well as other complex biological systems exhibit series of basic reproducible principles, which can be discovered through network analysis. This observation is encouraging because our primary objective is to discover such principles in temporal PDN. The network science methods were also used in study of cardiovascular diseases [59]. In [13] it was proposed to leverage the knowledge about diseases interconnectedness to provide enhanced care for non-communicable diseases. The information richness and scientific value of electronic patient records (EPR) similar to the ones studied in our work was demonstrated in [76]. Roque et. al [76] proposed to mine the PDN obtained from Danish EPR and augment it with free-text word mining for a more precise patient's diagnosis. The theme of EPR mining continues in [49], however, different from prior works, authors here consider legal, ethical and technical challenges that prevent use of combined PDN and genetic data [21] in practice.

5.1 Materials and Methods

The temporal PDN was constructed from the National Inpatient Sample (NIS), a collections of software and databases that are part of the Healthcare Cost and Utilization Project (HCUP) established in United State, a federal-state-industry partnership study. The NIS is a publicly available dataset containing about 80 million all-payer inpatient admission records in the United States. The NIS dataset differs from Medicare based data studied in [40]- it is a more realistic patient sample. In [40] the PDN was based on a sample of Medicare patients, who are by definition 65 years or older. The NIS dataset, on which our study is based, does not have this restriction. It is a sample from all patient populations in the United States. The sample is drawn from more than 4,000 hospitals located in 45 States, comprising a stratified sample of approximately 20% of all American hospitals. The sampling date range spans from 1998 to 2010, and here we concentrate on the latest years. For this study we selected 10 disjunct stratum presented in Table 18. In Table 18 the first column is the sample's year, followed by the number of admissions for that year, then the

number of hospitals within the stratum, the geographical region of the United States where the hospitals are located, the size-type of the stratum's hospitals in terms of number of beds, the ownership type and the teaching-location type. The teaching-location specifies the type of area where hospitals are located: urban vs. rural, and whether hospitals allow admission privileges to medical students. The 10 stratified samples were selected randomly, but preference was given to the samples with higher numbers of admissions, and therefore none of the stratum listed in Table 18 have hospitals that are small. The NIS data can be spotty at times, as it is not guaranteed that any given hospital was reporting its admissions for every month within a sampled year. Therefore, another criteria of stratified group selection was assurance that all hospitals within the group were reporting admissions for every month of the group's sampled year.

Table 18: Ten stratified samples from nationwide hospitals admission data.

Stratum	Year	Admissions Count	# of Hospitals	Region	Size	Ownership	Hospitals' Type
#1	2009	350,866	21	South	Large	Private Non-Profit	UNT
#2	2009	633,470	18	South	Large	Government	UT
#3	2008	31,629	37	Northeast	Large	Public	R
#4	2009	42,528	56	Northeast	Medium	Private	R
#5	2008	41,580	54	Northeast	Medium	Private	R
#6	2008	90,509	52	Northeast	Medium	Government	UNT
#7	2008	28,351	40	Northeast	Medium	Public	R
#8	2008	54,901	26	Northeast	Large	Private Non-Profit	UNT
#9	2007	205,010	34	South	Medium	Private	R
#10	2008	176,873	34	South	Medium	Private	R

Year, admission count,number of hospitals, region and group's hospitals' characteristics for each stratum. Hospitals' Type abbreviations, UNT -"Urban Non-Teaching", UT-"Urban Teaching" and R-"Rural".

Every admission record has up to 25 diagnoses codes in the ICD-9-CM coding scheme (www.cdc.gov/ nchs/ icd/ icd9.htm) and up to 25 corresponding diagnoses codes in Clinical Classifications Software (CCS) coding structure (http://www.hcup-us.ahrq.gov/ toolssoftware/ ccs/ ccsfactsheet.jsp). The ICD-9-CM coding scheme is internationally accepted dis-

ease classification nomenclature containing approximately 12,000 diagnoses codes. The large number of diagnoses codes is explained by their use in insurance claims by health care providers, but having 12,000 codes poses a problem for statistical analysis and reporting. This is why the CCS scheme was introduced with only 253 codes. The CCS groups together closely related ICD-9 codes to clinically meaningful disease categories, which greatly simplifies analysis. In this work the PDNs are constructed using CCS codes.

The temporal PDN for each stratum listed in Table 18 was constructed as following. Within a stratum's year we iterated over twelve months of the admission records. For each month we built a static PDN, where nodes were diseases under the CCS classification scheme and weighted links between nodes were counts of patients diagnosed with diseases pairs at the time of admission. This method was repeated for each month, resulting in temporal PDN consisting of 12 discrete monthly steps. Our sampling population has 10 stratums (Table 18), therefore 10 temporal networks were created. Equation (48) is the formula defining link weight between diseases k and l during j's month of stratum i - it is count of admission records out of total n_j admissions where both diseases k and l were diagnosed:

$$W_{kl}^{i,j} = \sum^{n_j} \mathbb{I}(\exists k \vee l) \tag{48}$$

Figure 15 is a histogram of a typical static PDN for a single month, and other PDNs are very similar to the one presented in Figure 15. The X-axis is an instance count, or weight of co-morbid disease pairings (Equation 48) and the Y-axis is the number of pairs. Figure 15 suggests distribution not unlike a power law distribution as has been observed in the past in many other network types [8]. The histogram in Figure 15 is "magnified" and concentrated on "elbow" to preserve scaling. In analyzed PDN graph some diseases co-occur more than 13,000 times while about 4,000 disease-pairs have very low co-occurrence. A very small number of links in Figure 15 have disproportionately large weights and a large number of links have weights that are very small. We also note that the weighted graph constructed

using our method is fully saturated. This is not surprising, because our large sample size virtually guarantees that for almost any disease pair there will be at least one patient who was diagnosed with both of them. Our goal is to understand how the whole phenotype disease network evolves but we also have to recognize the fact that observed co-morbidity of any two diseases could be spurious. We are also less interested in strength of correlation between rare diseases, which will never have large weight. This is why we applied a 90th percentile threshold to all monthly networks. Within each network we discarded links with weights lesser than that network's 90th percentile weight. This operation imposes sparsity on the network and allows us to investigate the top 10% salient relationships. Finally, we normalized all link values. Within each monthly static network we divided link weights by admissions count for the whole month. Thus a single month PDN became a weighted static graph where diseases (nodes) are connected by weighted links, and links' weights are the prevalence rates of disease pairs. More formally, the prevalence rate of diseases k and l for j's month in stratum i is expressed by Equation 49:

$$W_{kl}^{i,j} = \frac{\sum_{q}^{n_j} \mathbb{I}(\exists k \vee l)}{n_j} \tag{49}$$

where the n_j in Equation (49) is the number of admissions during month j. The prevalence rate in Equation (49) is not absolutely true, because some patients could have been readmitted within j's month, thus skewing the rate. Due to the privacy concerns the NIS data does not provide patients' identification information, so it is impossible to determine whether the same person was readmitted twice within a single month. We assume that such patients are in small numbers, but we should note this assumption as a limitation of the research. Different from our work, in Hidalgo et. al [40], the diseases were linked by Relative Risk measure and ϕ-correlation. Both measures quantify statistical significance of the relationship and in our work we are more interested in the prevalence rate. The prevalence rate allows estimation of the proportion of the population affected by illnesses, whereas the measure of statistical strength of the relationship cannot provide this information.

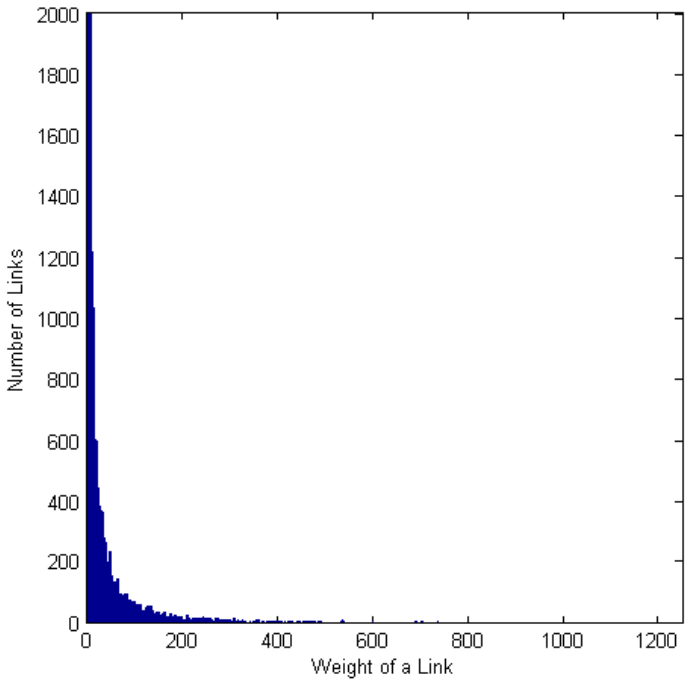

Figure 15: A static PDN histogram of unnormalized links' weights for one month. The X-axis is the number of admissions linking a pair of diseases, the Y-axis is the number of such links. The power law distribution is evident, there are few diseases that are disproportionately highly co-occurring.

5.2 Models For Temporal PDN

Our research objective is to define a statistical model capable of describing dynamics of the temporal PDN. The characteristics of a well fitted model will give us an insight into how temporal PDN evolves and the rules governing its evolution. One way to verify whether statistical model is a good fit is to apply it to predict what the network will look like at the next unobserved time step. In our study the next unobserved temporal period is a calendar month. We assume that an informative model should show advantage when predicting the future prevalence rates of co-morbid diseases. More formally, we are given the changing weighted non-directional temporal graph $\mathbf{W}^{T\times n\times n}$ of an invariant set of n nodes observed at discrete time intervals $t = 1 \dots T$ ($\mathbf{W}^{1,n\times n} \cdots \mathbf{W}^{T,n\times n}$), where for $i, j \in 1 \dots n$, $i < j$, $W^t_{ij} \in \mathbb{R}_{\geq 0}$, and by definition $W^t_{ii} = 0$. Our task is to infer what an undirected weighted

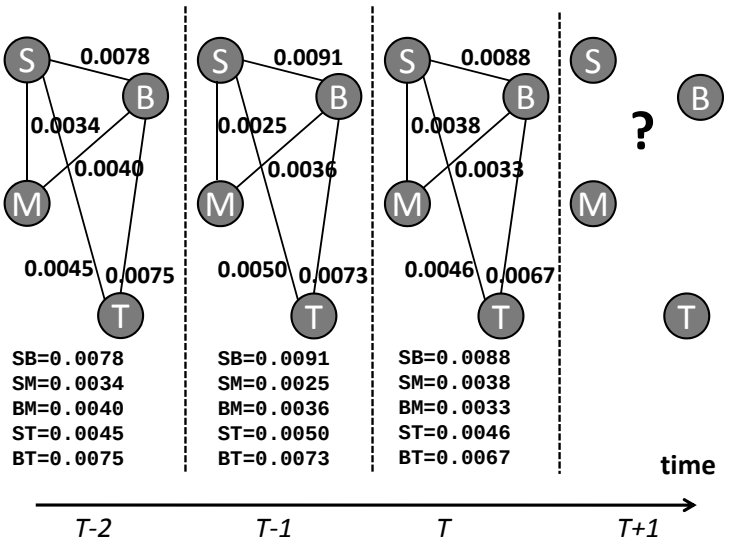

Figure 16: A conceptual representation of the prediction task addressed in this chapter. Given four diseases (S-"Septicemia", B-"Bacterial Infection", K-"Mycoses", T- "Thyroid Disease") and history of co-diagnosed prevalence rates in the previous 3 steps ($T-2 \ldots T$), the goal is to predict the prevalence rates at of disease co-occurrence future unobserved state $T+1$.

graph will look like at the next unobserved temporal step $\mathbf{W}^{T+1}$. The prediction task is schematically represented by Figure 16. In this problem setting we only use information about the links and the weights to predict the graph, and our model does not consider any information about the diseases.

5.2.1 Heuristics For Temporal PDN

In this section we discuss some simple heuristics and well-known models applicable to our study. One heuristic which can be used to predict a future link's weight (prevalence rate of co-occurring diseases i and j) is to take its last known value:

$$W_{ij}^{T+1} = W_{ij}^{T} \tag{50}$$

another method is to take the mean of its n last known weights:

$$W_{ij}^{T+1} = \frac{1}{n} \sum_{t=T-n+1}^{T} W_{ij}^{t} \tag{51}$$

Murata and Moriyasu [63] developed "weighted" variants of Common Neighbor [64] and Adamic-Adar [1] algorithms for prediction of unweighted binary links. The weighted version of the Common Neighbor algorithm [63] is defined as:

$$W_{ij} = \sum_{k \in \Gamma(i) \cap \Gamma(j)} \frac{W_{ik} + W_{jk}}{2} \tag{52}$$

The "weighted" Adamic-Adar is expressed as:

$$W_{ij} = \sum_{k \in \Gamma(i) \cap \Gamma(j)} \frac{W_{ik} + W_{jk}}{2} \times \frac{1}{log(\sum_{k' \in \Gamma(k)} W_{k'k})} \tag{53}$$

The function Γ in Equations (52) and (53) is a neighborhood function, and it returns immediately connected neighbors of its node-argument. Intuitively, weighted Common Neighbor averages the weight of all paths connecting nodes i and j through immediate neighbors. The weighted Adamic-Adar does the same but moderates predicted weight by log sum of each common neighbor's weighted links. Equations (52) and (53) do not reference time index because both formulas are designed for static weighted networks. To make temporal PDN suitable for calculations by both functions, it has to be "flattened", i.e. we have to have its static invariant. The new static network can be created from a temporal set by taking the historical mean of each link (Equation 51) so that the network can be treated by weighted Common Neighbor and Adamic-Adar algorithms.

5.2.2 Statistical Models

The modeling and prediction of the weighted graphs was covered in the network science literature extensively. However, surprisingly, none of the previously published works addressed the prediction problem we have reviewed. Here we provide a brief overview of the

literature and how it relates to us. In [96] authors proposed a weighted network evolution model for the traffic flow. The model in [96] uses randomized heuristics to simulate the progress of the weighted graph. Here, the network topography features, as well as outdegree distributions are used as controlled variables for evaluation of artificial graphs. It is not clear however how the model parameters can be learned from the real life network so this approach cannot be used for prediction of the graph's future state. De Sa and Prudencio [23] proposed a supervised prediction technique for weighted graphs, but the graph in [23] is static and in our work it is dynamic. There are other works that address prediction of the future step in temporal networks but the networks in those works are binary [25, 46].

5.2.3 Proposed Model

We attempt to model the the dynamics of temporal PDN by proposing Weighted Temporal Exponential Random Graph Model (WTERGM), which is based on the Temporal Exponential Random Graph Model (tERGM) [37]. Our proposed model is a generalized solution for prediction of the temporal network. It can predict not only the presence or absence of a link, but also its strength expressed by the link's real valued weight. WTERGM is a Markovian model; it assumes that weighted network matrix $\mathbf{W}^t$ is conditionally independent of prior temporal observations $\mathbf{W}^1 \cdots \mathbf{W}^{t-2}$ given its prior observed state $\mathbf{W}^{t-1}$. This assumption is expressed by Equations (54):

$$P(\boldsymbol{W}^t|\boldsymbol{W}^{t-1}, \boldsymbol{W}^{t-2} \cdots \boldsymbol{W}^1) = P(\boldsymbol{W}^t|\boldsymbol{W}^{t-1}) \tag{54}$$

Therefore the joint distribution of WTERGM is expressed by:

$$P(\boldsymbol{W}^{1:T}|\boldsymbol{\theta}) = P(\boldsymbol{W}^1) \prod_{t=2}^{T} P(\boldsymbol{W}^t|\boldsymbol{W}^{t-1}, \boldsymbol{\theta}) \tag{55}$$

We define the transition probability distribution as

$$P(\boldsymbol{W}^t|\boldsymbol{W}^{t-1},\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta},\boldsymbol{W}^{t-1})} exp\{\boldsymbol{\theta}^{'}\boldsymbol{\psi}(\boldsymbol{W}^t,\boldsymbol{W}^{t-1})\}\mathbb{N}(\boldsymbol{W}^t|\boldsymbol{W}^{1:T}) \tag{56}$$

The function $\mathbb{N}$ in the Equation (56) is regularization prior and its purpose is to restrain prediction of the link to be reasonably within the range of its past observed values. We define $\mathbb{N}$ as a Gaussian multivariate distribution with its $1 \times \frac{n(n-1)}{2}$ vector mean $\boldsymbol{\mu}$ set to the average of each link's observed history and $\frac{n(n-1)}{2} \times \frac{n(n-1)}{2}$ independent covariance matrix with only non-zero entries on its diagonal, where each entry σ_m for $m = 1 \ldots \frac{n(n-1)}{2}$ is historical standard deviation of m's link. The weighted temporal graph in this work is non-directional, hence we can have $\frac{n(n-1)}{2}$ possible links (n is number of nodes). The independence assumption of the prior's variance is deliberate and it serves two purposes, it keeps the model parsimonious and improves its scalability.

The transitional probability statistic's vector $\boldsymbol{\psi}$ from (56) consists of results of l functions in the $\mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \to \mathbb{R}$ form. In our model we define the total of $l = 3$ functions, called *density*, *stability* and *variance*:

$$\psi_D(\boldsymbol{W}^t,\boldsymbol{W}^{t-1}) = \frac{1}{n}\sum_{ij}^{n} W_{ij}^t \tag{57}$$

$$\psi_S(\boldsymbol{W}^t,\boldsymbol{W}^{t-1}) = \sum_{i<j}^{n} \mathbb{I}(W_{ij}^t - W_{ij}^{t-1} < \sigma_{ij}) \tag{58}$$

$$\psi_V(\boldsymbol{W}^t,\boldsymbol{W}^{t-1}) = \sum_{i<j}^{n} \mathbb{I}(|W_{ij}^t - \mu_{ij}| < \sigma_{ij}) \tag{59}$$

The statistic ψ_D controls the density of the weighted graph, so it is useful when the weighted graph is sparse. The statistic ψ_S controls temporal stability of the transitional step, counting the number of links which during transition from time step $t-1$ to t changed less than link's historical standard deviation. The ψ_V models variance of the network $\mathbf{W}^t$, it counts links which did not deviate from their historical mean μ_{ij} by value greater than

link's historical standard deviation σ_{ij}.

The learning of the parameter weight vector $\boldsymbol{\theta}$ is done through iterative EM MCMC process and it is well documented in [37, 36, 68, 82]. The inner loop of estimation procedure iterates over network's temporal history and draws collection of network samples [82] from each time step. The partial derivatives then computed based on collected samples and model's weights $\boldsymbol{\theta}$ are updated using Newton-Raphson optimization in the outer loop. The iteration continues until updates to the parameter $\boldsymbol{\theta}$ become sufficiently small. The sampling process of the inner loop is Metropolis-Hastings sampling and here we provide its description. Given time-step t, the vector parameter $\boldsymbol{\theta}$ and last known PDN, $\mathbf{W}^t$, we start a Metropolis-Hastings sampling process to draw samples from the posterior distribution $P(\mathbf{W}^{t+1}|\mathbf{W}^t,\boldsymbol{\theta})$. We randomly initialize starting graph $\mathbf{W}^{(1),t+1}$, then the Metropolis-Hastings sampling algorithm circles through the graph's $\mathbf{W}^{(1),t+1}$ links stochastically updating it, defining the Markov chain process, which asymptomatically approximates random graph distribution specified by $\mathbf{W}^t$ and $\boldsymbol{\theta}$. At each stochastic update we consider only a single link, and this is why the graphs $\mathbf{W}^{(u),t+1}$ and $\mathbf{W}^{(u+1),t+1}$ at update step u differ from each other by value of a single link. The update works as follows: at update step u we define graph $\mathbf{W}^{ij,(u),t+1}$, which is an exact copy of the graph $\mathbf{W}^{(u-1),t+1}$ from the previous update step except the weight of link between i and j, which was randomly selected for the update. We set the value of this link $W_{ij}^{ij,(u),t+1}$ to the proposed new value:

$$W_{ij}^{ij,(u),t+1} = max(0, W_{ij}^{(u-1),t+1} + \beta\mathcal{N}(0, \sigma_{ij})) \tag{60}$$

In Equation (60) we use Gaussian as proposed jump distribution with 0 mean and σ_{ij} historical variance of ij's link. The prevalence rate by definition is always positive, so if the proposed value becomes negative we set it to 0. The parameter β regulates the magnitude of a jump and is customarily set within range $(0 \dots 1]$. The decision to accept or reject jump to the $\mathbf{W}^{ij,(u),t+1}$ is computed using acceptance ratio formula given by Equation (61):

$$\alpha = \frac{exp\{\boldsymbol{\theta}'\boldsymbol{\psi}(\boldsymbol{W}^{ij,(u),t+1}, \boldsymbol{W}^{t})\}\mathcal{N}(W_{ij}^{ij,(u),t+1}|\mu_{ij}, \sigma_{ij})}{exp\{\boldsymbol{\theta}'\boldsymbol{\psi}(\boldsymbol{W}^{(u-1),t+1}, \boldsymbol{W}^{t})\}\mathcal{N}(W_{ij}^{(u-1),t+1}|\mu_{ij}, \sigma_{ij})} \tag{61}$$

In Equation (61) we moderate the chance of accepting proposed value by incorporating Gaussian prior with link's parameters μ_{ij} and σ_{ij}. This technique (Equation 56) imposes restriction upon the drawn values so they will be within an interval reasonably close to links' observed means.

5.3 Results

Our hypothesis is that WTERGM's statistics are capable of capturing the processes governing the evolution of temporal PDN. If our model can predict the future unobserved state of a weighted graph better than commonly known baselines, it would suggest that the model's statistics are informative of the temporal network. The experiments were setup as follows, we separated temporal PDN within each stratum into the training and test parts. The training part consisted of 11 monthly time steps and the test part was a graph for the last (12th) month. The prediction error of each algorithm was measured by mean of absolute error:

$$e = \frac{1}{n}\sum_{i<j}^{n} |\hat{W}_{ij} - W^{12}ij| \tag{62}$$

Here, the $\hat{\mathbf{W}}$ is the prediction, $\mathbf{W}^{12}$ is 12th month graph we are predicting and n is number of nodes.

We then applied the WTERGM model and each baseline algorithm to predict 12th month of each stratum from Table 18. The prediction results are presented in Table 19. In conducted experiments the "Average" and "Last Step" methods performed far better than more advanced weighted "Adamic Adar" and "Common Neighbor" methods. This is a telling sign that the common neighbor paradigm often cited in network science literature is not applicable in weighted temporal networks. Specifically, the WTERGM model was better than the next best baseline 8 times out of 10.

Table 19: Mean absolute error of predicting disease co-occurrence at hospitals scaled by $\times 10^3$ of WTERGM and baselines models for the 12th month graph based on previous 11 monthly time-steps in each stratum.

Stratum	Average	Last Step	Adamic Adar	Common Neighbor	WETRGM
#1	8.92	<u>7.59</u>	505.19	3811.25	**7.18**
#2	6.07	<u>5.80</u>	427.51	3075.64	**5.28**
#3	<u>10.70</u>	12.36	186.37	1399.33	**10.13**
#4	**11.44**	13.54	234.52	1764.92	<u>12.18</u>
#5	<u>9.55</u>	11.10	204.34	1519.11	**9.08**
#6	<u>7.67</u>	8.11	196.03	1374.81	**7.61**
#7	<u>9.86</u>	10.60	149.42	1098.61	**9.28**
#8	13.06	<u>12.45</u>	287.88	2216.71	**12.14**
#9	7.20	7.22	301.26	2175.59	**6.80**
#10	**7.52**	<u>8.72</u>	344.18	2523.23	**7.52**

Empirical evaluation of various algorithms' accuracy predicting weighted PDN graph of the last 12th month in each stratum. The best predictor for the stratum is in bold, the second best predictor is underlined. The prediction error is expressed by Equation (62).

In Figure 17 we closely examine the distribution of prediction error when using "Average", "Last Step" and "WERGM" algorithm. We do not include the error distributions of weighted "Common Neighbor" and "Adamic Adar" because their prediction errors are out of scale with other algorithms. Each boxplot in Figure 17 represents the distribution of absolute error predicting link weights for the 12th month graph in stratum #1. Here we can see that the interquartile range and 1.5 interquartile range (denoted by whiskers) is tighter for WTERGM. Also there are less outliers, corresponding to larger errors, in WTERGM's error distribution.

5.4 Discussion

The empirical results presented in Table 19 suggest that WTERGM model and its statistics are descriptive of PDN's evolution. We explore this concept further by running an additional experiment. We run all prediction algorithms in round robin fashion multiple times and predict the 12th time step of a randomly picked stratum. In Figures 18,19 and 20 each data point represents a weighted graph prediction. The X-axis of the plots is the predic-

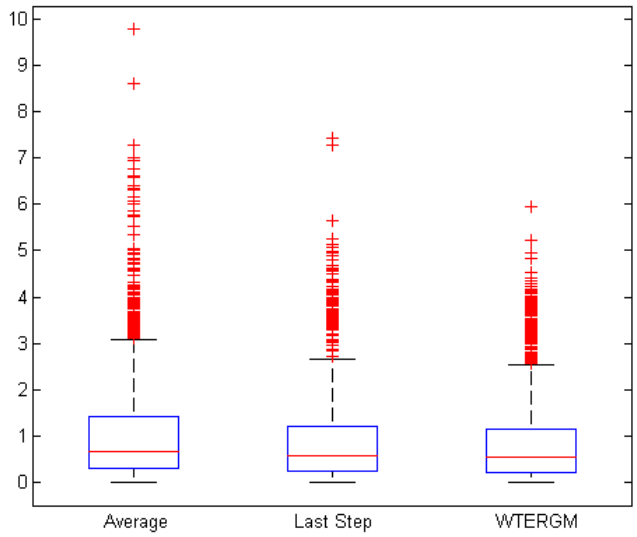

Figure 17: "Average", "Last Step" and "WTERGM" prediction error distributions scaled by $\times 10^3$ for the 12th month disease co-occurrence in stratum #1.

tion error (62) and the Y-axis in Figure 18 is the absolute difference between the *stability*, Equations (57), of the predicted network and the true *stability* value of the network being predicted. Correspondingly, in Figure 19 we compare *density* statistics, Equation (58), and in Figure 20 *variance*, Equation (59). There is an obvious linear trend between the predictor's error and the distance between each individual statistics. The larger the prediction error, the further the distance is between the true statistics and the statistics of the predicted graph. Therefore Figures 18,19 and 20 suggest that WTERGM statistics are informative of observed temporal PDN.

The statistics ψ_D, Equation (57), controls the density of the weighted graph. Our experiments suggest that it remains fairly constant as PDN changes over time. This indicates that energy of the PDN is fairly static and is time invariant, reducing likelihood that PDN's average prevalence rate will spike or drop off significantly over time. While any prevalence rate can change with time because of variations in admissions and/or patients "traveling" from one disease to another one, the averaged prevalence rate of the whole graph will not

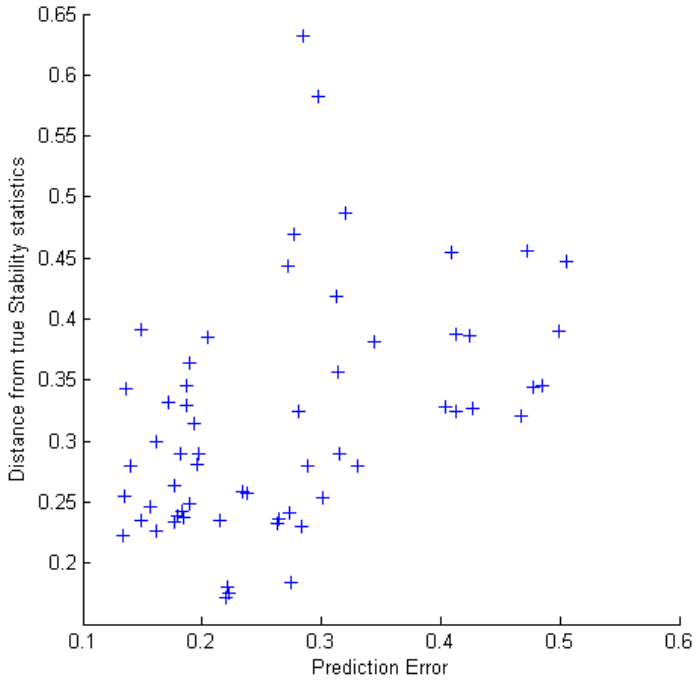

Figure 18: WTERGM informative *stability* statistics. Each data point is one weighted graph prediction, where the X-axis is prediction error scaled by $\times 10^3$, Equation (62), while the Y-axis is the absolute distance between *stability* statistics value, Equation (58), of the predicted network and ground truth.

fluctuate much. The next statistics ψ_S, Equation (58), counts links that remain stable over time. Based on our empirical evaluations we posit that ψ_S is a value that is also not prone to change. One possible explanation is that ψ_S captures chronic diseases that are almost always co-diagnosed and are slow-moving such that we do not expect their prevalence rates to change in monthly snapshots granularity. Lastly, statistic ψ_V, Equation (59), identifies prevalence rates which significantly deviate from their observed historical values. The fact that ψ_V works well as part of the model probably can be explained in a manner opposite to ψ_S. Where ψ_S captures relationships between slowly ongoing chronic diseases, while the ψ_V statistic accounts for diseases with fast-changing relationships. It is important to note that ψ_S and ψ_V are not absolute complements of each other. The ψ_S capture temporal stability, i.e. from time step to time step, the ψ_V is a "too hot", "too cold" indicator at the

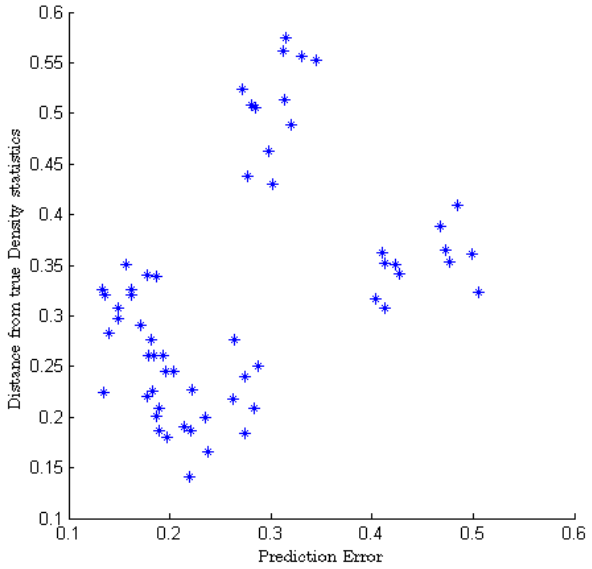

Figure 19: WTERGM informative *density* statistics. Each data point is one weighted graph prediction, where the X-axis is prediction error scaled by $\times 10^3$, Equation (62), while the Y-axis is the absolute distance between *density* statistics value, Equation (57), of the predicted network and ground truth.

present moment and it does not depend as much on past link weight.

The WTERGM's statistics provide helpful insight on selected PDN's aspects. The initial results are encouraging and other statistics should be explored in hope of providing more information on laws governing temporal PDN. The statistics presented in this chapter are only a small subset of network statistics known to today's science. We have tried other network statistics as well. For example, we attempted to integrate a triad census expressed as:

$$\psi_T(\boldsymbol{W}^t, \boldsymbol{W}^{t-1}) = \sum_{i<j<k}^{n} \mathbb{I}(W_{ij}^t > \mu_{ij} \wedge W_{jk}^t > \mu_{jk} \wedge W_{ik}^t > \mu_{ik}) \tag{63}$$

counts "hot" triads, fully connected diseases which prevalence rates are much larger than historical means. However, we were not successful in applying it to our model, and these statistics were not predictive of PDN's future behavior. The number of "hot" triads was changing dramatically over time and it could be just a random spurious factor. This runs contrary to the reported importance of transitive relationships in social networks [73, 75],

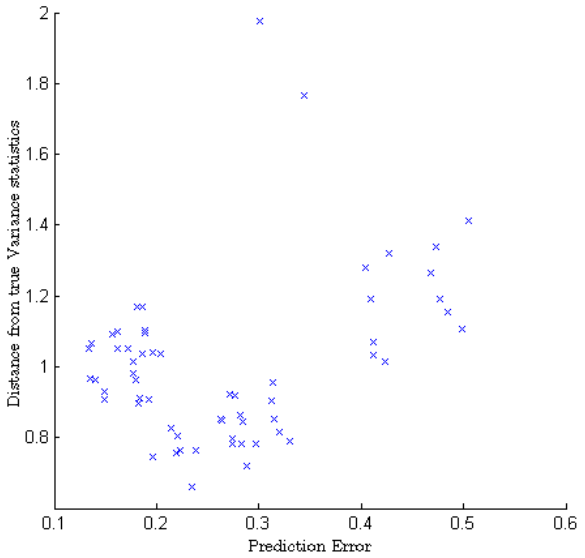

Figure 20: WTERGM informative *variance* statistics. Each data point is one weighted graph prediction, where the X-axis is prediction error scaled by $\times 10^3$, Equation (62), while the Y-axis is the absolute distance between *variance* statistics value, Equation (59), of the predicted network and ground truth.

thus highlighting the difference between temporal PDN's mechanism and the mechanism of a "flat" social network.

The study of PDN's dynamics is important, but a promising practical aspect is also present in this study. It is conceivable that a statistical model which can predict diseases prevalence rates of the immediate future can be used in practice to augment resource planning by health care providers. A model could predict changes in levels of diagnosis of many diseases in a hospital or group of hospitals. Such information could be incorporated into the decision-making process on allocation of valuable health care resources.

CHAPTER 6

CONCLUSION

The networks science research is exciting and certainly a booming research area. In this work we reviewed and studied problems from multiple domains: social networks, social science and medicine. There are, however, many more problems that are waiting solution in many other areas which can use network analysis. We demonstrated the versatility and robustness of log-linear approach through multiple sets of experiments. The empirical evaluations suggest validity of log linear approaches. There is however, is room for improvement, the scalability to large networks,the kind that can be found in Internet computer applications, is often concern. Another issue is that log linear models are not always deterministic and often are hard to implement. At the same time log linear model's assumptions expressed in model's statistics or model's structure itself are fairly easy to interpret. For example, if the statistic measuring temporal stability of the network helps prediction it tells us a lot about the network itself such that it is governed by fairly slowly evolving mechanism. Another benefit is "pluggability" of many models we had presented. A researcher can simply design a new statistic she think is expressive of the network's mechanics and can simply add such statistic to the model. Such plug-and-play feature allows researchers to do a rapid iterations in testing of various hypothesis about the network.

REFERENCES CITED

[1] L.A Adamic and E. Adar, 'Friends and neighbors on the web', *Social Networks*, **25**, (2001).

[2] A. Ahmed and E.P. Xing, 'Recovering time-varying networks of dependencies in social and biological studies.', *Proceedings of the National Academy of Sciences of the United States of America*, **106**, (2009).

[3] EM Airoldi, DS Choi, and PJ Wolfe, 'Confidence sets for network structure.', *Stat Anal Data Min*, **4**(5), 461–469, (2011).

[4] H. Akaike, 'A new look at the statistical model identification', *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, **19**, (1974).

[5] GC Alexander and AR Sehgal, 'Why hemodialysis patients fail to complete the transplantation process', *Am J Kidney Dis*, **37**(2), 321–328, (2001).

[6] JZ Ayanian, PD Cleary, JS Weissman, and AM Epstein, 'The effect of patients' preferences on racial differences in access to renal transplantation', *N Engl J Med*, **341**(22), 1661–1669, (1999).

[7] B.W. Bader and T.G. Kolda, 'Efficient matlab computations with sparse and factored tensors', *SIAM Journal on Scientific Computing*, **30**, 205–231, (2007).

[8] A. Barabasi and R. Albert, 'Emergence of scaling in random networks', *Science*, **286**, (1999).

[9] A. Bartal, E. Sasson, and G. Ravid, 'Predicting links in social networks using text mining and sna', *Social Network Analysis and Mining, International Conference on Advances in Social Network Analysis and Mining*, **30**, (2009).

[10] HR Bernard, P Killworth, D Kronenfeld, and L Sailer, 'The problem of informant accuracy: The validity of retrospective data.', *Annu Rev Anthropol*, **13**(1), 495–517, (1984).

[11] Julian Besag, 'Spatial interaction and the statistical analysis of lattice systems', *Journal of the Royal Statistical Society. Series B (Methodological)*, **36**, 192–236, (1974).

[12] S.P. Borgatti and J.L. Molina, 'Ethical and strategic issues in organizational social network analysis', *Journal of Applied Behavioral Science*, **39**(3), 337–349, (2003).

[13] Jean Bousquet, Josep Anto, and Peter Sterk, 'Systems medicine and integrated care to combat chronic noncommunicable diseases', *Genome Medicine*, **3**, 1–12, (2011).

[14] G.E.P. Box and G.M. Jenkins, *Time series analysis; forecasting and control*, Holden-Day, San Francisco, CA, USA, 1st edn., 1970.

[15] A.P. Bradley, 'The use of the area under the roc curve in the evaluation of machine learning algorithms', *Pattern Recognition*, **30**, (1997).

[16] T Browne, 'The relationship between social networks and pathways to kidney transplant parity: Evidence from black americans in chicago.', *Soc Sci Med*, **73**(5), 663–667, (2011).

[17] R.S. Burt, 'A note on missing network data in the general social survey', *Social Networks*, (1987).

[18] P.J. Carrington, J. Scott, and S. Wasserman, *Models and methods in social network analysis*, Cambridge University Press, New York, NY, USA, 1st edn., 2005.

[19] J. Carroll and Jih-Jie Chang, 'Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition', *Psychometrika*, **35**, 283–319, (1970).

[20] Hui Chang, Bei-Bei Su, Yue-Ping Zhou, and Da-Ren He, 'Assortativity and act degree distribution of some collaboration networks', *Physica A: Statistical Mechanics and its Applications*, **383**(2), 687–702, (2007).

[21] Darcy A. Davis and Nitesh V. Chawla, 'Exploring and exploiting disease interactions from multi-relational gene and phenotype networks', *PLoS ONE*, **6**(7), e22670, (07 2011).

[22] W. de Nooy, A. Mrvar, and V. Batagelj, *Exploratory social network analysis with pajek*, Cambridge University Press, New York, NY, USA, 1st edn., 2005.

[23] H.R. de Sa and R.B.C. Prudencio, 'Supervised link prediction in weighted networks', in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pp. 2281 –2288, (August 2011).

[24] Peter B. DeOreo, 'Hemodialysis patient-assessed functional health status predicts continued survival, hospitalization, and dialysis-attendance compliance', *American Journal of Kidney Diseases*, **30**(2), 204–212, (1997).

[25] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar, 'Temporal link prediction using matrix and tensor factorizations', *ACM Transactions on Knowledge Discovery from Data*, **5**, 1–10, (February 2011).

[26] S. Dynes, P.A. Gloor, R. Laubacher, and Y. Zhao, 'Temporal visualization and analysis of social networks', in *Proceedings of the North American Association for Computational Social and Organizational Science Conference*. North American Association for Computational Social and Organizational Science, (2004).

[27] E Edgington, *Randomization Tests (Statistics: A Series of Textbooks and Monographs).*, CRC Press, Calgary, Alberta, Canada, 3rd edn., 1995.

[28] AM Epstein, JZ Ayanian, JH Keogh, SJ Noonan, N Armistead, PD Cleary, JS Weissman, JA David-Kasdan, D Carlson, J Fuller, D Marsh, and RM Conti, 'Racial disparities in access to renal transplantation - clinically appropriate or due to underuse or overuse?', *N Engl J Med*, **343**(21), 1537–1544, (2000).

[29] RA Fisher, 'On the interpretation of χ^2 from contingency tables, and the calculation of p.', *J R Stat Soc Series*, **85**(1), 87–94, (1922).

[30] C Fraley and D Percivaly, 'Model-averaged l1 regularization using markov chain monte carlo model composition', *Technical Report No. 541, Department of Statistics, University of Washington*, (2011).

[31] O Frank and D Strauss, 'Markov graphs.', *J Am Stat Assoc*, **81**(395), 832–842, (1986).

[32] Y Freund, 'The alternating decision tree learning algorithm', in *Machine Learning: Proceedings of the Sixteenth International Conference*, pp. 124–133, (1999).

[33] J Friedman, T Hastie, and R Tibshirani, 'Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors).', *Ann Stat*, **28**(2), 337–407, (2000).

[34] A Gillespie, H Hammer, J Lee, C Nnewihe, J Gordon, and P Silva, 'Lack of listing status awareness: Results of a single-center survey of hemodialysis patients.', *Am J Transplant*, **11**(7), 1522–1526, (2011).

[35] F. Guo, S. Hanneke, W. Fu, and E.P. Xing, 'Recovering temporally rewiring networks: a model-based approach', in *Proceedings of the 24th international conference on Machine learning*, New York, NY, USA, (2007). Association for Computing Machinery.

[36] S. Hanneke, W. Fu, and E.P. Xing, 'Discrete temporal models of social networks', *Electronic Journal of Statistics*, **4**, (2010).

[37] S. Hanneke and E. Xing, 'Discrete temporal models of social networks', in *Proceedings of the International Conference on Machine Learning Workshop on Statistical Network Analysis*, New York, NY, USA, (2006). Springer-Verlag.

[38] W.K. Hastings, 'Monte carlo sampling methods using markov chains and their applications', *Biometrika*, **57**, (1970).

[39] F. Heider, 'Attitudes and cognitive organization', *Journal of Psychology*, **21**, (1946).

[40] Csar A. Hidalgo, Nicholas Blumm, Albert-Lszl Barabsi, and Nicholas A. Christakis, 'A dynamic network approach for the study of human phenotypes', *PLoS Comput Biol*, **5**(4), (04 2009).

[41] G.W. Hill, 'Turrialba, social systems and the introduction of change', *Rural Sociology*, **19**, (1954).

[42] PD Hoff, 'Bilinear mixed-effects models for dyadic data.', *J Am Stat Assoc*, **100**(469), 286–295, (2005).

[43] PD Hoff, 'Multiplicative latent factor models for description and prediction of social networks.', *Computational and Mathematical Organization Theory*, **15**(4), 261–272, (2009).

[44] PD Hoff, AE Raftery, and MS Handcock, 'Latent space approaches to social network analysis.', *J Am Stat Assoc*, **97**(460), 1090–1098, (2002).

[45] JL Holley, C McCauley, B Doherty, L Stackiewicz, and JP Johnson, 'Patients' views in the choice of renal transplant', *Kidney Int*, **49**(2), 494–498, (1996).

[46] Z. Huang and D.K.J. Lin, 'The time-series link prediction problem with applications in communication surveillance', *Institute for Operations Research and the Management Sciences Journal on Computing*, **21**, (2009).

[47] M. Huisman, 'Imputation of missing network data: Some simple procedures', *Journal of Social Structure*, **10**, (2009).

[48] M. Huisman and C. Steglich, 'Treatment of non-response in longitudinal network studies', *Social Networks*, **30**(4), 297 – 308, (2008).

[49] PB Jensen, LJ Jensen, and S Brunak, 'Mining electronic health records: towards better research applications and clinical care', *Nature Reviews Genetics*, **13**, 395–405, (2012).

[50] L. Katz, 'A new status index derived from sociometric analysis', *Psychometrika*, **18**, (1953).

[51] S. Kim and S. Imoto, S.and Miyano, 'Dynamic bayesian network and nonparametric regression model for inferring gene networks from time series microarray data', **75**, (2003).

[52] AC Klassen, AG Hall, B Saksvig, B Curbow, and DK Klassen, 'Relationship between patients' perceptions of disadvantage and discrimination and listing for kidney transplantation.', *Am J Public Health*, **92**(5), 811–817, (2002).

[53] Johan H Koskinen, Garry L Robins, and Philippa E Pattison, 'Analysing exponential random graph (p*) models with missing data using bayesian data augmentation', *Statistical Methodology*, **7**(3), 366–384, (2010).

[54] Gueorgi Kossinets, 'Effects of missing data in social networks', *Social Networks*, **28**, 247–268, (2006).

[55] M. Lahiri and T.Y. Berger-Wolf, 'Structure prediction in temporal networks using frequent subgraphs', in *Proceedings of the Institute of Electrical and Electronics Engineers Symposium on Computational Intelligence and Data Mining*, Los Alamitos, CA, USA, (2007). Institute of Electrical and Electronics Engineers Press.

[56] D.A. Levin, Y. Peres, and E.L. Wilmer, *Markov chains and mixing times*, American Mathematical Society, New York, NY, USA, 1st edn., 2008.

[57] L. Li, J. McCann, N.S. Pollard, and C. Faloutsos, 'Dynammo: mining and summarization of coevolving sequences with missing values', in *Proc. 15th ACM SIGKDD*, (2009).

[58] Y Liu, M Liang, Y Zhou, Y He, Y Hao, M Song, C Yu, H Liu, Z Liu, and T Jiang, 'Disrupted small-world networks in schizophrenia.', *Brain*, **131**(4), 945–961, (2008).

[59] Aldons J. Lusis and James N. Weiss, 'Cardiovascular networks', *Circulation*, **121**(1), 157–170, (2010).

[60] DJ Marchette and CE Priebe, 'Predicting unobserved links in incompletely observed networks', *Comput Stat Data Anal*, **52**(3), 1373–1386, (2008).

[61] L. Michell and A. Amos, 'Girls, pecking order and smoking', *Social Science and Medicine*, **44**, (1997).

[62] J. Moody, 'Matrix methods for calculating the triad census', *Social Networks*, **20**, (1998).

[63] Tsuyoshi Murata and Sakiko Moriyasu, 'Link prediction of social networks based on weighted proximity measures', in *Web Intelligence, IEEE/WIC/ACM International Conference on*, pp. 85 –88, (2007).

[64] M.E.J. Newman, 'Clustering and preferential attachment in growing networks', *Physical Review E*, **64**(2), (2001).

[65] NIH, *US Renal Data System, USRDS 2010 Annual Data Report: Atlas of Chronic Kidney Disease and End-Stage Renal Disease in the United States*, Bethesda, MD, 2010.

[66] D.L. Nowell and J. Kleinberg, 'The link prediction problem for social networks', in *Proceedings of the 12th international conference on Information and knowledge management*, New York, NY, USA, (2003). Association for Computing Machinery.

[67] V Ouzienko, A Gillespie, H Hammer, T Browne, and Z Obradovic, 'In review: Perceived bias in the renal allocation system predicts mortality in black american hemodialysis patients', *Transplantation*, (2012).

[68] V Ouzienko, Y Guo, and Z Obradovic, 'A decoupled exponential random graph model for prediction of structure and attributes in temporal social networks', *Stat Anal Data Min*, **4**(5), 470–486, (2011).

[69] J Park and IW Sandberg, 'Universal approximation using radial-basis-function networks.', *Neural Comput*, **3**(2), 246–257, (1991).

[70] M.L. Pearson and L. Michell, 'Smoke rings: social network analysis of friendship groups, smoking and drug-taking', *Drugs: education, prevention and policy*, **7**, (2000).

[71] A. Popescul, R. Popescul, and L.H. Ungar, 'Statistical relational learning for link prediction', in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, (2003).

[72] G. Robins, T. Snijders, P. Wang, and M. Handcock, 'Recent developments in exponential random graph (p*) models for social networks', *Social Networks*, **29**, (2006).

[73] Garry Robins, Peter Elliott, and Philippa Pattison, 'Network models for social selection processes', *Social Networks*, **23**(1), 1 – 30, (2001).

[74] Garry Robins, Philippa Pattison, and Jodie Woolcock, 'Missing data in networks: exponential random graph (p*) models for networks with non-respondents', *Social Networks*, **26**(3), 257–283, (2004).

[75] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher, 'An introduction to exponential random graph (p*) models for social networks', *Social Networks*, **29**(2), 173 – 191, (2007).

[76] Francisco S. Roque, Peter B. Jensen, and Henriette Schmock, 'Using electronic patient records to discover disease correlations and stratify patient cohorts', *PLoS Comput Biol*, **7**, e1002141, (08 2011).

[77] Ruslan Salakhutdinov and Andriy Mnih, 'Probabilistic matrix factorization', in *Advances in Neural Information Processing Systems 20*, eds., J.C. Platt, D. Koller, Y. Singer, and S. Roweis, 1257–1264, MIT Press, Cambridge, MA, (2008).

[78] ES Schaeffner, J Mehta, and WC Winkelmayer, 'Educational level as a determinant of access to and outcomes after kidney transplantation in the united states', *Am J Kidney Dis*, **51**(5), 811–818, (2008).

[79] JL Schafer, 'Multiple imputation: a primer', *Statistical Methods in Medical Research*, **8**(1), 3, (1999).

[80] Y Slinin, RN Foley, and AJ Collins, 'Calcium, phosphorus, parathyroid hormone, and cardiovascular disease in hemodialysis patients: The usrds waves 1, 3, and 4 study.', *J Am Soc Nephrol*, **16**(6), 1788–1793, (2005).

[81] T. Snijders, C. Steglich, and G. van de Bunt, 'Introduction to stochastic actor-based models for network dynamics', *Social Networks*, **32**, (2009).

[82] T.A.B Snijders, 'Markov chain monte carlo estimation of exponential random graph models', *Journal of Social Structure*, **3**, (2002).

[83] T.A.B. Snijders, 'Models for longitudinal network data', in *Models and Methods in SNA*, (2005).

[84] Tom A. B. Snijders, Philippa E. Pattison, Garry L. Robins, and Mark S. Handcock, 'New specifications for exponential random graph models', *Sociological Methodology*, **36**, 99–153(55), (December 2006).

[85] Christian Steglich, Tom A. B. Snijders, and Michael Pearson, 'Dynamic networks and behavior: separating selection from influence', *Sociological Methodology*, **40**(1), 329–393, (2010).

[86] K Steinhaeuser, NV Chawla, and AR Ganguly, 'Complex networks as a unified framework for descriptive analysis and predictive modeling in climate science.', *Stat Anal Data Min*, **4**(5), 497–511, (2011).

[87] Alexey Stomakhin, Martin B Short, and Andrea L Bertozzi, 'Reconstruction of missing data in social networks based on temporal patterns of interactions', *Inverse Problems*, **27**(11), (2011).

[88] D. Stork and W.D. Richards, 'Nonrespondents in communication network studies', *Group and Organization Management*, (1992).

[89] M Thamer, W Hwang, NE Fink, JH Sadler, EB Bass, AS Levey, R Brookmeyer, and NR Powe, 'U.s. nephrologists' attitudes towards renal transplantation: results from a national survey', *Transplantation*, **71**(2), 281–288, (2001).

[90] T. Tylenda, R. Angelova, and S. Bedathur, 'Towards time-aware link prediction in evolving social networks', in *Proceedings of the The 3rd workshop of Social Network Mining and Analysis - The 13th Association for Computing Machinery International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, (2009). Association for Computing Machinery.

[91] ML Unruh, IV Evans, NE Fink, NR Powe, and KB Meyer, 'Skipped treatments, markers of nutritional nonadherence, and survival among incident hemodialysis patients.', *Am J Kidney Dis*, **46**(6), 1107–1116, (2005).

[92] Marc Vidal, MichaelE. Cusick, and Albert-Lszl Barabsi, 'Interactome networks and human disease', *Cell*, **144**(6), 986 – 998, (2011).

[93] S Wasserman and P Pattison, 'Logit models and logistic regressions for social networks: I. an introduction to markov graphs and p* .', *Psychometrika*, **61**(3), 401–425, (1996).

[94] L. Xiong, X. Chen, T. K. Huang, J. Schneider, and J. G. Carbonell, 'Temporal collaborative filtering with bayesian probabilistic tensor factorization', in *SIAM Data Mining 2010 (SDM 10)*, (2010).

[95] K Yeates, N Wiebe, J Gill, C Sima, D Schaubel, D Holland, B Hemmelgarn, and M Tonelli, 'Similar outcomes among black and white renal allograft recipients', *Clin J Am Soc Nephrol*, **20**(1), 172–179, (2009).

[96] Jian-Feng Zheng and Zi-You Gao, 'A weighted network evolution with traffic flow', *Physica A: Statistical Mechanics and its Applications*, **387**(24), 6177 – 6182, (2008).

[97] M Zrinyi, M Juhasz, J Balla, E Katona, T Ben, G Kakuk, and D Pall, 'Dietary self-efficacy: determinant of compliance behaviours and biochemical outcomes in haemodialysis patients.', *Nephrol Dial Transplant*, **18**(9), 1869–1873, (2003).