2018

A Model for Seasonal Dynamic Networks

Jace D. Robinson

Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation

https://corescholar.libraries.wright.edu/etd_all/1920

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
A Model for Seasonal Dynamic Networks

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

by

Jace D. Robinson
B.S.C.S., Wright State University, 2016

2018
Wright State University
Wright State University
GRADUATE SCHOOL

April 9, 2018


________________________________________
Derek Doran, Ph.D.
Thesis Director

________________________________________
Mateen M. Rizki, Ph.D.
Chair, Department of
Computer Science and Engineering

Committee on
Final Examination

________________________________________
Derek Doran, Ph.D.

________________________________________
Tanvi Banerjee, Ph.D.

________________________________________
Fred Garber, Ph.D.

________________________________________
Barry Milligan, Ph.D.
Interim Dean of the Graduate School
Sociotechnological and geospatial processes exhibit time varying structure that make insight discovery challenging. This paper presents statistical model of systems with seasonal dynamics, modeled as a dynamic network, to address this challenge. It assumes the probability of edge formations depend on a type assigned to incident nodes and the current time. Time dependencies are modeled by unique seasonal processes. The model is studied on several synthetic and real datasets. Superior fidelity of this model on seasonal datasets compared to existing network models, while being able to remain equally accurate for networks with randomly changing structure, is shown. The model is found to be twice as accurate at predicting future edge counts over competing models on New York City taxi trips, U.S. airline flights, and email communication within the Enron company. An anomaly detection use case for the model is shown for NYC traffic dynamics and email communications between Enron employees.
Contents

1 Introduction 1

2 Preliminaries 5
   2.1 Random Networks ................................................. 5
   2.2 Seasonal Time Series ............................................. 9

3 Related Work 10

4 Methodology 14
   4.1 Model Specification ................................................ 14
   4.2 Model Fitting ....................................................... 17
      4.2.1 State Space Model ......................................... 17
      4.2.2 Kalman Filtering and Smoothing ............................ 21
      4.2.3 Maximum Likelihood Estimation ............................. 24
      4.2.4 Expectation-Maximization .................................. 24

5 Results 28
   5.1 Synthetic Experiments ............................................ 29
   5.2 Evaluation on Real Data ................................. 31
      5.2.1 Edge Count Prediction .................................... 32
   5.3 Anomaly Detection Use Case ................................. 33
      5.3.1 NYC Taxi Trips ............................................ 37
      5.3.2 Enron E-mails ............................................. 38
   5.4 Theoretical Analysis ............................................. 41

6 Conclusion 45

Bibliography 49
## List of Figures

1.1 Example of seasonality in dynamic networks. In this three class network, a latent seasonal process determines a probability of edge formation at times $t_1$ and $t_2$, subject to both process and measurement noise. A different process (colored plates on the right) affect probabilities for edges connecting unique pairs of node types. The thesis presents and evaluates a statistical model codifying these ideas, which may be useful in comparison, prediction, and anomaly detection tasks on dynamic complex systems. 3

2.1 Two adjacency matrices generated from the same parameters of a four class directed SBM. The vertices for the rows and columns are sorted by vertex label, to group the blocks together. The cell is shaded blue if an edge exists, unshaded otherwise. 8

4.1 Example of transition noise, causing the estimated seasonal pattern in red to deviate from the original seasonal curve in green. 15

4.2 Example of observation noise, where the blue observations have high variance, but remain along the seasonal pattern in green. 16

4.3 Graphical model of the SDSBM in state space model form. The darker circles are of variables assumed given, while the white circles are inferred. 20

4.4 Comparison of the inferred seasonality (posterior distribution) from Kalman Smoother compared to actual seasonality governing connections between nodes of type $x$ and $y$. The underlying seasonality follows a simple sine wave that is accurately recovered from noisy observations. 23

5.1 MSE comparisons of each model to recover the true SBM parameters over increasing noise levels and seasonal patterns of increasing amplitude. 30

5.2 Seasonal time series of edge counts in each dataset. 32
5.3 In panels (a), (b), and (c) are the edge counts prediction results for each model on the three datasets. In panel (d), the absolute error (AE) on NYC Taxis across time is presented, providing qualitative evidence for unmodeled structure in the DSBM and DSBMEWMA. In panel (e), the autocorrelations (AC) for the absolute errors on NYC Taxis with 95% confidence intervals (CI) are calculated, providing further quantitative evidence for unmodeled structure in the DSBM and DSBMEWMA.

5.4 Image of GEONET software tool with NYC Taxi dataset.

5.5 Cumulative loglikelihood over all blocks per time step for the New York City dynamic network. Plot best viewed digitally.

5.6 Edge counts in the Queens to Queens block with prominent spikes and drops in value on January 1st, February 9th, and March 14th.

5.7 A node with anomalous degree in the Bronx on May 14th, 2017.

5.8 Time series of edge counts between blocks of the Enron e-mail network with anomalies.

5.9 This is a plot of data generated and fit from the typical state space model with only the transition noise and observation noise, without the additional measurement noise parameter. The time series of edge counts data is in black, forecasted mean fit in red, and 95% confidence bounds in green.

5.10 This is a plot of data generated and fit from the new state space model with all three of the noise terms. The time series of edge counts data is in black, forecasted mean fit in red, and 95% confidence bounds in green.
List of Tables

5.1 Overview of real datasets ........................................ 32
5.2 Loglikelihood of 4 most anomalous time steps, with the three most anomalous blocks per time step shown in bold. ................. 40
Acknowledgment

I would like to thank my advisor Dr. Doran for giving me the opportunity and pleasure to have him as a role model these past two years. Before joining his team, I was lost, unmotivated, and unsure what to do for a career. Through witnessing his passion, skill, and joy of research, he convinced me to pursue a career in research.

I would like to thank my graduate school friends for their advice and entertainment over these past two years. Thank you to Matt Piekenbrock for the many, many long arguments about machine learning, statistics, and research philosophy. Thank you to Cogan, Fan, Garrett, Mahdieh, Ning, and Reza constant comedic relief. Thank you to Jameson for valuable help in programming the GEONET software tool and preparing for the sponsor meeting.

Thank you to my family for their infinite support and confidence despite not understanding what I do for a living.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation’s I/UCRC Center for Surveillance Research. This work is supported by industry and government partners at the National Science Foundations I/UCRC Center for Surveillance Research and the Air Force Research Laboratory.
In honor of my father Mark Robinson.
1

Introduction

Relational data are prevalent in our daily lives through many technological, sociological, and biological systems. The Internet consists of over a billion website connected through a system of links. In 2017 Twitter has over 328 million monthly users actively communicating with each other\(^1\). The human brain consists of 86 billion neurons firing and interacting with each other to enable complex human behavior\(^2\). These are three examples of massive networks. A network is simply a mathematical way to describe relational data in the form of vertices and edges. Classical tools from statistics and machine learning like linear regression or Gaussian distributions cannot be directly applied to network data. A set of mathematical tools and techniques were developed in the past few decades to model this data. This led to the growing field of Network Science.

Networks that are static, meaning their vertices, edges, and structure are assumed fixed with time, have been well studied, with results collected and presented in textbooks\(^{[21]}\). In reality, the assumptions of a network being static is incorrect. The Internet, Twitter, and even human biology are rapidly changing with time. Due to this and the increasing availability of large, dynamic network datasets, active research areas have moved to the study of dynamic networks, where at least one feature of the network is changing with

\(^1\)Statistic from Statista
\(^2\)Statistic from Nature
time. Depending on the property of the network that is changing, an array of new dynamic network modeling techniques need to be developed. This thesis develops one of those techniques.

Many complex systems exhibit regular, time dependent, seasonal patterns. For example, human movement patterns are driven by the time of day [27], and vehicle traffic densities exhibit predictable increases at certain hours causing rush hours and decreases at night [17]. This same ‘seasonal’, time dependent effect occurs when monitoring network bandwidth usage [35], when counting the number of clicks per day on a web page [8], or when tracking mobile traffic levels [30]. We look to bring the notion of seasonality to statistical network modeling with a new kind of dynamic stochastic block model (DSBM). A DSBM asserts that system components (vertices) are grouped into several types, and the probability of observing a component relation or interaction (edges) are determined by the types of the incident nodes and time. Different kinds of DSBM consider different assumptions about the network formation process, but none consider seasonality [33, 32, 18, 34, 28, 29]. These models stand to under fit temporal patterns in seasonal data.

A conceptual overview of the model we propose is given in Figure 1.1. It fuses structural time series (plates on the right of Figure 1.1) with a generative network model. We call this a seasonal DSBM (SDSBM). This model was first introduced in [25].

By learning a model of the seasonal patterns in dynamic network data, several applied problems can be approached. One of the most common applications is for prediction. Given data of the dynamic network up to a certain time, can we predict the structure of the network in future time steps? One specific variant of this problem is known as link prediction in dynamic networks. In social networks the ability to predict whether two people are to become friends creates a recommendation system. In networks of vehicle traffic, traffic patterns can be optimizing by predicting and avoiding high density locations. In web traffic, predicting future server usage can allow more cost efficient resource management. In this thesis we demonstrate the superior predictive capabilities of the new model on several
Figure 1.1: Example of seasonality in dynamic networks. In this three class network, a latent seasonal process determines a probability of edge formation at times $t_1$ and $t_2$, subject to both process and measurement noise. A different process (colored plates on the right) affect probabilities for edges connecting unique pairs of node types. The thesis presents and evaluates a statistical model codifying these ideas, which may be useful in comparison, prediction, and anomaly detection tasks on dynamic complex systems.

Another use case of the proposed SDSBM is for anomaly detection. A task can be to find a small amount of meaningful or interesting information out of a massive dataset, analogous to finding a needle in a haystack. Some examples of this include spam detection in emails, bank fraud in transaction histories, or crimes from surveillance data. A specific application of interest is for anomaly detection in massive surveillance data over a geospace. Given a dynamic network representation of movement in a geospace, where vertices are locations, and edges are movement between locations, the SDSBM is able separate the normal moments of time from the abnormal. We demonstrate this anomaly detection capability in geospatial data of a six month dataset of millions of New York City taxi trips.

Chapter Overview

The rest of the thesis is organized as follows.
Chapter 2: Preliminaries  describes the fundamental concepts vital to the understand of the model of random networks and seasonal time series.

Chapter 3: Related Work  discusses many of the approaches for modeling and anomaly detection on dynamic networks.

Chapter 3: Methodology  defines the generative process, inference procedure to recover the latent seasonal patterns driving the stochastic blockmodel, and learning algorithm to fit model parameters given dynamic network data.

Chapter 4: Results  performs several synthetic and real data experiments to demonstrate the superiority of the SDSBM over existing models on seasonal datasets and applicability of it to prediction and anomaly detection tasks.

Chapter 5: Conclusions  summarizes the contributions from this thesis and proposed several directions for future work.
2

Preliminaries

In this chapter, the necessary background material to understand the main concepts not covered in the methodology chapter of the thesis are provided.

2.1 Random Networks

Just as continuous data like heights of females can be modeled as following a distribution, the same ideas can be applied to networks. Individual networks $G_1, G_2, ..., G_n$ can be thought of as data from a distribution of all possible networks defined by some model. One of the simplest and most well know of these models is the Erdős-Rényi (ER) [26],[21]. The model itself was created by Solomonoff and Rapoport, but was popularized by series of papers by Paul Erdős and Alfréd Rényi. This model is defined as the $G\{n,p\}$, where $n$ is the fixed number of vertices in the network, and $p$ is the probability of edge formation between any two vertices. This defines a distribution of networks with $n$ vertices for all possible realization of edges. With the distribution, we can evaluate the probability of a realized network. More formally:

$$Pr(G_i|n,p) = p^m(1-p)^{\binom{n}{2} - m}$$
where $m$ is the number of edges is network $G_i$, and $\binom{n}{2}$ is the number of possible edges in a directed network. The method provides a simple example of how to define a distribution over networks, but is too simplistic to reflect networks in the world.

A generalization of the ER model is the Stochastic Blockmodel (SBM) originally developed by Holland [12]. Instead of all edges following the identical probability distribution, there are several distributions based on the *labels* of the adjacent vertices. More precisely a SBM is defined by (using notation from [5]):

1. $n$ is the number of vertices in $G$.

2. $A$ an adjacency representation of graph $G$ where element $a_{ij} = 1$ if there is an edge between vertex $i$ and vertex $j$, zero otherwise. Within this thesis all graphs are assumed binary and directed unless otherwise specified.

3. $k$ is the number of possible vertex labels or classes (terms used interchangeably in this thesis)

4. A $n \times k$ position assignment matrix $Z$, where element $z_{ij} = 1$ if vertex $i$ has label $j$, and zero otherwise. In this thesis we assume each vertex belongs to exactly one position, although variants allowing partial assignment to each class (soft assignment) exist [1].

5. A $k \times k$ block density matrix $E$.

This allows use to define the distribution over networks in a similar form to the ER model, but with the added vertex labels:

$$Pr(A|Z,E) = \prod_{i \sim j} \prod_{k,l} (E_{kl}^{a_{ij}}(1-E_{kl})^{1-a_{ij}})^{z_{ik}z_{jl}}$$
A nice feature of the SBM is that it is a generative network model. This means given the model parameters $Z$ and $E$, adjacency $A$ can be created. That is for all $i$ and $j$, find the only $k$ and $l$ for which:

$$z_{ik}z_{jl} = 1$$

and then assign

$$A_{ij} \sim Bernoulli(E_{kl})$$

There are several important properties of the SBM that should be understood. The reasoning for the name blockmodel becomes more apparent when you look at the adjacency matrix shown as shown in figure 2.1. The example SBM has four class labels, producing sixteen independent blocks, communities or subgraphs (terms used interchangeably in this thesis), as shown by the sixteen squares. All edges within a block are independently and identically distributed (IID). As each block will be independent of each other in our formulation, and hence the equations presented in chapter 4 are independently fit for each block.

Completing the sampling from equation 2.1 will create a static adjacency representation of network, where neither the vertices nor edges have any time dependency. In the chapters 3 and 4, we overview several dynamic variants, which instead of producing a single adjacency matrix $A$, produce a sequence of discrete time dependent adjacency matrices $\mathbb{A} = \{A_1, A_2, \ldots, A_T\}$, where either the edges or vertices change with time.
Figure 2.1: Two adjacency matrices generated from the same parameters of a four class directed SBM. The vertices for the rows and columns are sorted by vertex label, to group the blocks together. The cell is shaded blue if an edge exists, unshaded otherwise.
2.2 Seasonal Time Series

The main contribution of this thesis is combining the idea of the SBM with a seasonal time series model. When looking at a dynamic network given as a sequence of adjacency matrices \( A = \{A_1, A_2, \ldots, A_T\} \), with given vertex labels, it is useful to look at the edge counts per block. By the assumptions of the SBM, all edges within a block are considered IID, and therefore knowing the number of edges per block and the maximum number of possible edges within a block contains the same amount of information to the model as the adjacency matrix. The sequence of adjacencies can then be represented as a sequence of edge counts per block, which is what it often plotted such as on the plates in the conceptual figure 1.1. Seasonality in a time series is a regular pattern of changes that repeats over \( d \) time periods. As we are dealing with discrete time steps, a simple deterministic seasonal pattern for example (with \( d = 3 \)) could be \( \{x_1, x_2, x_3, \ldots\} = \{1, 2, 3, 1, 2, 3, 1, \ldots\} \), etc...

With real data, the observed seasonal data is assumed to be noisy realizations of the underlying seasonal pattern, so values could be \( \{0.8, 2.1, 3.0, 1.1, \ldots\} \), where the mean of the underlying seasonal distribution is the original deterministic path.
3

Related Work

The original stochastic block model (SBM) [12] has been a successful statistical random network. It builds a framework to intuitively capture the idea that a network can be comprised of communities, where each community has its own probability of edge formation. This approach allows for the use of optimization and inference on data to determine community structure in observed networks. It has been theoretically explored [14] and applied [7] extensively on static networks. A survey of SBMs is available [19].

It is only natural with the success of the original SBM that dynamic variants would be created. Each new variant of the model encodes a different set of assumptions to solve slightly different problems. The most common motivation is the problem of dynamic community detection. In [33], the authors assume a dynamic stochastic blockmodel (DSBM), where the probability of edge formation follows a random walk in time. Using an extended Kalman Filter model, a likelihood of the entire network given community labels was defined. An optimization routine was developed to infer community labels by maximizing the likelihood as the community labels were iteratively changed. The authors were able to show their approach achieved similar accuracy in recovering communities and predicting edges to the competing models on several synthetic and real datasets, but was faster. This work both closely aligns and inspired the ideas of this thesis, and is compared to in
the evaluation. In [32], the same author extends the work to removes the hidden Markov assumptions on edge-level dynamics. The author uses the idea that whether or not an edge existed at the previous time step will have a large influence on how likely it is to appear in the next. A similar extended Kalman filter plus local search was described to infer communities from dynamic network data. In both papers [33] and [32], the author defines two variants of the model, one where vertex labels are assumed given, and another where the vertex labels are inferred. In the evaluation section, we only compare our model against the former variant, as we assume the labels can be provided in the data.

In [34], instead of modeling changing probability of edge formation, the authors allow vertices to randomly change types through time following a random walk. Using a Bayesian framework, posterior distributions of the model parameters are derived. The authors demonstrated the superiority of their model for community detection over many previous static and dynamic approaches.

A possible problem with the assumptions of both SBMs and DSBMs, are the generated networks do not reflect the degree distribution found in real data. The DSBM will have degree distributions that appear approximately binomial, while real data has heavy tail distributions (e.g. the DSBM does not model ‘hub’ nodes of high degree). In [14], the authors developed a degree corrected variant of the SBM to alleviate this issue. In [28], the authors bring degree correction to the dynamic variant. Using statistical process monitoring techniques, the authors setup a change detection problem, where at some time $t < t^*$, the network has a set of model parameters, but then from time $t > t^*$, these model parameters changed. They demonstrate the applicability of their methodology on networks of voting patterns in the U.S. Senate. Xing et al. consider a mixed membership SBM with dynamic tomography, allowing vertices to carry different types through time [29]. Ho et al. extend the previous work on dynamic mixed membership SBM, this time allowing vertices to evolve in a cluster, as opposed to individually, improving the inferential power [11]. The original inspiration for the seasonal time series was from [4], where the authors used sea-
sonal time series model for anomaly detection on the counts of pedestrian traffic in front of a stationary camera.

Existing approaches are built on the idea that some property of the observed networks would evolve \textit{randomly} with time, rather than \textit{seasonally} or \textit{periodically}. The SDSBM is designed to have enough model complexity to accurately recover seasonal patterns in data, which a number of real geospatial, social, and technological complex systems exhibit. We show how the SDSBM models the dynamics of such systems with higher fidelity compared to the current art in Chapter 5.

Beyond SBMs is a broad area of research into anomaly detection over dynamic networks. An excellent survey paper on the topic is available [23]. The survey defines a taxonomy of different anomaly detection approaches, for anomalous vertices, anomalous edges, anomalous subgraphs, or change point detection. To quantify what is ‘anomalous’, researchers have explored community detection, compression, decomposition, distance and probabilistic metrics. The SDSBM defines a probabilistic anomaly detector for abnormal subgraphs and a form of change point detection. For dynamic network anomaly detection specifically over geospatial data, there has been recent work from [31] and [22]. In the former paper, the authors developed a dynamic network anomaly detection algorithm using higher-order networks. Most existing models rely on the Markov assumption that the current time step is predictable given information from the previous time step. By incorporating more knowledge from multiple previous time steps in a higher order method, more accurate predictions can be developed. The usefulness of their model is demonstrated on geospatial data of taxis in Porto Portugal. The authors were able to detection change points in the system during the \textit{Burning of the Ribbons} festival. In the latter paper the authors developed a model for the detection of the evolution of community structure using a dynamic weighted SBM. Instead of the simple SBM where edges are assumed to be binary and following Bernoulli distributions, it is possible to model weighted edges following Poisson, exponential, or normal distributions. Using a nonparametric hypothesis test to determine
if all networks in a sequence come from the same distribution, the change in a network of New York City is detection from Hurricane Sandy.

Not every approach for dynamic networks uses a statistical model. Some of the most effective approaches in anomaly detection have been based on evaluating the similarity between consecutive network snapshots. In [16], the ideas of graph similarity are used to define DELTACON, a principled technique for anomaly detection in massive graphs. The authors focus on a set of axioms to ensure the algorithm achieves desirable properties. Some of the axioms include notions of edge importance, weight awareness, edge-submodularity and focus awareness. The effectiveness of the algorithm is shown on a dynamic network formulation of the Enron email dataset, detecting key events such when Kevin Lay resigned from the board, or with Skilling, Fastow, and Kopper appeared at Congress. In the evaluation chapter, we will discuss some of the similarities and differences of the techniques and anomalies found from [22] and [16] to those found by the SDSBM.
Methodology

4.1 Model Specification

We assume that time is discrete, with the current time $t \in [1, 2, \ldots, T]$ representing a time period of some resolution. We also assume the node types (also referred to as labels or classes) are given. Each pair of types $a$ and $b$ define a block of edges. For each block $(a, b)$, we consider an independent structural time series with a bias $m_t^{(a,b)}$ establishing an anchor for values of the time series and a seasonal offset $s_t^{(a,b)}$ that shifts the bias by the seasonality position. The process at time $t$, denoted $c_t^{(a,b)}$, is

$$c_t^{(a,b)} = m_t^{(a,b)} + s_t^{(a,b)}$$ \hspace{1cm} (4.1)

with bias $m_t^{(a,b)}$ described by

$$m_t^{(a,b)} = m_{t-1}^{(a,b)} + \delta m_t^{(a,b)}$$ \hspace{1cm} (4.2)

where $\delta m_t^{(a,b)} \sim \mathcal{N}(0, q_m^{(a,b)})$. The set of seasonal offsets are stored in a vector $s_t^{(a,b)} = (s_1^{(a,b)}, s_2^{(a,b)}, \ldots, s_d^{(a,b)})$ having $d$ components. $d$ reflects either the length or the resolution of
Figure 4.1: Example of transition noise, causing the estimated seasonal pattern in red to deviate from the original seasonal curve in green.

A seasonal process (e.g., $d = 60$ to model per minute changes over a process that cycles per hour) and is assumed to be provided by the user of the model. The components are:

$$s_t^{(a,b)} = - \sum_{i=1}^{d-1} s_{t-i}^{(a,b)} + \delta_s^{(a,b)}$$

(4.3)

where $\delta_s^{(a,b)} \sim \mathcal{N}(0, q_s^{(a,b)})$. This form enforces a zero-sum constraint to increase identifiability [20]. It should be emphasized that $q_m^{(a,b)}$ and $q_s^{(a,b)}$ control the transition noise, determining how the underlying seasonal process changes across time. For low noise levels, the seasonal process will follow a rigid pattern that repeats, but at high noise levels will become random, with the seasonal pattern changing significantly every time step. A visual example of this noise is presented in Figure 4.1.
The seasonal process governs edge formations between nodes of type $a$ and $b$ by a random variable $p_t^{(a,b)} \in [0, 1]$ specifying the expected density of edges spanning node types $(a,b)$ at time $t$:

$$p_t^{(a,b)} = c_t^{(a,b)} + \epsilon p_t^{(a,b)}$$

(4.4)

where $\epsilon p_t^{(a,b)} \sim \mathcal{N}(0, b^{(a,b)})$ control a portion of the observation noise. While the transition noise determines how the process will change over time, the observation noise determines the uncertainty at a single time step. For low values, the expected density will closely resemble the underlying seasonal pattern, but for high values will appear random. In figure 4.2 an example of the observation noise is shown. Additional exploration and discussion of these noise parameters on synthetic data are completed in chapter 5.4.

Now we define an adjacency matrix $A_t$, where $[A_t]_{ij} = 1$ if there exists an edge.
between nodes \( i \) and \( j \) at time \( t \) and \( [A_t]_{ij} = 0 \) otherwise. Denote \( A_t^{(a,b)} \) as the submatrix of \( A_t \) only containing the rows and columns representing type \( a \) and type \( b \) nodes. Then \( A_t^{(a,b)} \) is defined by the random variable:

\[
[A_t^{(a,b)}]_{ij} \sim Bernoulli(p_t^{(a,b)}) \tag{4.5}
\]

Repeating this process for all blocks \((a, b)\) and time steps \( t \) will generate a dynamic network \( \mathcal{A} = \{A_1, A_2, ..., A_T\} \). The randomness in the edge generation additionally contributes to the observation noise.

### 4.2 Model Fitting

We now describe an inference procedure to fit the latent seasonal processes of the model to an observed \( \mathcal{A} \) with known vertex types. The task is to estimate a posterior distribution on each \( m_t^{(a,b)} \) and \( s_t^{(a,b)} \) for each pair of node types \((a, b)\). Kalman Filters [13] and Kalman Smoothers [24] in combination with numerical maximum likelihood estimators are appropriate tools for this task, but requires transforming the generative model into a state-space model (SSM).

#### 4.2.1 State Space Model

A SSM is a time series model with hidden state \( x_t \) and observed \( w_t \) variables [20], respectively. A SSM creates observations at time \( t \) by two linear models: An observation model

\[
w_t = h x_t + \epsilon_t \tag{4.6}
\]

and a transition model

\[
x_t = G x_{t-1} + \Delta_t \tag{4.7}
\]
Observations $w_t$ are generated by a transformation of the output (defined by $h$) of the underlying transition model. The transition model describes transformations within a hidden state space where transitions from time $t - 1$ to time $t$ are defined by the matrix $G$. Observations and transitions are to be subject to time dependent random noise modeled by gaussian distributions $\epsilon_t \sim N(0, r_t)$ and $\Delta_t \sim N(0, Q_t)$ with $r_t$ and $Q_t$ controlling the amount of observation and transition noise, respectively. If parameters $\theta_t = \{h, G, r_t, Q_t\}$ are known, a Kalman Filter can derive the posterior of the hidden state at time $t$ given all past observations $Pr(x_t|w_1, w_2, ..., w_t; \theta_t)$.

Now we transform the model specification into a state space to define the transition model $x_t = Gx_{t-1} + \Delta_t$ [6, 4]. We assume edges of different node types $(a,b)$ are independent of each other, so we formulate the inference in terms of a pair $(a,b)$. The explicit reference of blocks $(a,b)$ is dropped for notational simplicity, as the full inference process is independently completed over all blocks. We transform Equations 4.2 and 4.3 to define the hidden state variable $x_t$ and state transition $G$. The hidden state will be composed of the bias and vector of seasonal offsets as a $d \times 1$ seasonal state vector $x_t = \begin{bmatrix} m_t & s_t & s_{t-1} & \cdots & s_{t-d+2} \end{bmatrix}^T$ where $^T$ is the transpose operator. Note that all the seasonal offsets from $s$ are maintained in the state for a given $t$, with the $d^{th}$ seasonal offset implicitly defined based on the zero-sum constraint. To perform a state transition from time $t - 1$ to time $t$ we define $G$ as the $d \times d$ matrix:

$$G = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & -1 & \cdots & -1 & -1 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

(4.8)

Multiplication of the first row of $G$ by $x_t$ yields Equation 4.2 without noise being added,
as only the bias term is updated. Multiplication of the second row of $G$ by $x_t$ will update a single seasonal offset as shown in Equation 4.3. The remaining rows of $G$ serve to permute the remaining seasonal offsets, such that each offset $s_i$ is updated after a full period of $d$ time steps. Each time step will update the most current seasonal offset, and shift the remaining offsets down one index in the state vector. Next we define the $d \times 1$ transition noise vector $\Delta_t = [\delta_{m_t}, \delta_{s_t}, 0, \ldots, 0]^T$ where the first element is the bias noise in Equation 4.2, the second element is seasonal noise in Equation 4.3, and the remaining elements are all 0 as there is no additional noise for permuting the seasonal offsets. These noise values are sampled from a zero mean gaussian with $d \times d$ covariance matrix $Q = \text{diag}[q_m, q_s, 0, \ldots, 0]$. Assuming the bias noise $\delta_{m_t}$ and seasonal offset noise $\delta_{s_t}$ are independent, the off-diagonal elements of $Q$ are zero. We assume $Q$ is stationary and thus drop the dependence on $t$.

Our next task is to transform Equations 4.4 and 4.5 into the observation model $w_t = hx_t + \epsilon_t$. To do this, we first define a block size $n_{(a,b)}$ as the maximum number of edges in block $(a, b)$ of a directed network, so if there are $|a|$ nodes of type $a$ and $|b|$ nodes of type $b$ then define:

\[
  n_{(a,b)} = \begin{cases} 
    a = b & |a|(|a| - 1) \\
    a \neq b & 2|a||b|
  \end{cases}
\]  

(4.9)

Also define the random variable $e_t \sim \text{Binomial}(p_t, n)$ as the number of formed edges in block $(a, b)$ at time $t$, where $p_t$ is the expected edge density as determined by Equation 4.4. $e_t$ is simply a more mathematically convenient way to define the edge generation process from Equation 4.5 and does not change the overall model. We then approximate this binomial distribution with a gaussian distribution of fixed variance as:

\[
e_t = np_t + \epsilon_{e_t}
\]  

(4.10)

where $\epsilon_{e_t} \sim \mathcal{N}(0, \sigma)$. This approximation allows for efficient use of the Kalman Filter as a
Figure 4.3: Graphical model of the SDSBM in state space model form. The darker circles are of variables assumed given, while the white circles are inferred.

linear gaussian system.

Now that we have transformed Equation 4.5 to an approximately gaussian formulation in Equation 4.10, we can define the parameters of the observation model \( w_t \) and \( h \). We define the observed variable \( w_t \) as the number of formed edges \( e_t \). To create \( h \), define a transformation which takes as input a seasonal state vector \( x_t \) and produces as output the number of formed edges \( w_t \). Combining the operations of Equations 4.4 and 4.10 lets us define: 

\[
    h = \begin{bmatrix} n & n & 0 & \ldots & 0 \end{bmatrix}. \quad w_t = hx_t
\]

can thus be seen as summing the bias \( m_t \) and seasonal offset \( s_t \) following Equation 4.4, and multiplying by \( n \) to match the repeated Bernoulli trials from Equation 4.10. The observation noise \( \epsilon_t \) is created by combining \( \epsilon_{pt} \) from Equation 4.4 and \( \epsilon_{et} \) from Equation 4.10. \( \epsilon_t \) is sampled from a zero mean stationary gaussian with variance \( r = u + n^2b \).

We have now transformed the generative procedure to the suitable SSM to allow efficient inference of the latent seasonality via the Kalman Filter and Kalman Smoother when \( \theta = \{ h, G, r, Q \} \) are known. A graphical model representation of the SDSBM is provided in figure 4.3
4.2.2 Kalman Filtering and Smoothing

The Kalman Filter is an algorithm that uses a sequence of noisy observations over time to produce an estimate of the distribution of underlying latent state variables. Starting with an initial guess of the latent state distribution \( P_r(X_{1|0}; \theta) = N(\mu_{1|0}, \Sigma_{1|0}) \), the algorithm recursively applies a prediction and update step using the state space formulation to estimate the posterior state distribution \( P_r(x_t|w_1, w_2, \ldots, w_t; \theta) \) for all \( t \). For gaussian models, the Kalman Filter provides an optimal solution to recover the distribution of the latent state.

The notations \( t|t-1 \) and \( t|t \) to be used emphasize the estimates at time \( t \) are created using observations up to time \( t-1 \) and time \( t \) respectively. We also use \( w_{1:t} \) as a compact representation of \( \{w_1, w_2, \ldots, w_t\} \).

Given observations up to time \( t-1 \), the prediction step uses the assumed transition model to forecast the expected state distribution one step ahead at time \( t \): \( P_r(x_t|w_{1:t-1}; \theta) = N(\mu_{t|t-1}, \Sigma_{t|t-1}) \),

\[
\mu_{t|t-1} = G\mu_{t-1|t-1} \tag{4.11}
\]

\[
\Sigma_{t|t-1} = G\Sigma_{t-1|t-1}G^T + Q \tag{4.12}
\]

where \( \mu_{t|t-1} = \mathbb{E}[x_t|w_{1:t-1}] \) and \( \Sigma_{t|t-1} = \mathbb{E}[(x_t - \mu_{t|t-1})(x_t - \mu_{t|t-1})^T|w_{1:t-1}] \). This updates the state mean \( \mu_{t|t-1} \) by permuting the seasonal offsets in \( x_{t|t-1} \), and updates the state variance \( \Sigma_{t|t-1} \) by adding uncertainty from \( Q \).

With the new observation \( w_t \), the update step combines the predicted state \( P_r(x_t|w_{1:t-1}; \theta) \) with the observation to refine the state estimate: \( P_r(x_t|w_{1:t}; \theta) = N(\mu_{t|t}, \Sigma_{t|t}) \),

\[
\mu_{t|t} = \mu_{t|t-1} + k_t(w_t - h\mu_{t|t-1}) \tag{4.13}
\]

\[
\Sigma_{t|t} = (I - k_t h)\Sigma_{t|t-1} \tag{4.14}
\]

\[
k_t = \Sigma_{t|t-1}\Sigma_{t|t-1}h^T(h\Sigma_{t|t-1}\Sigma_{t|t-1}h^T + r)^{-1}, \text{ where } \mu_t := \mathbb{E}[x_t|w_{1:t}] \text{ and } \Sigma_{t|t} := \mathbb{E}[(x_t - \mu_{t|t})(x_t - \mu_{t|t})^T]\]
\( \mu_{t|t} \)\( ^T \)\( |w_{1:t} \), and \( k_t \) is known as the Kalman gain matrix. To build intuition on these equations, we examine Equation 4.13 closely, which calculates the new state mean as a combination of the predicted state mean plus a correction factor from the residual between the actual observation \( w_t \) and the predicted expected observation \( h\mu_{t|t-1} \), scaled by the Kalman gain \( k_t \). If \( |k_t| \) is large, which will occur when \( r \) approaches zero, then the new state mean will be largely estimated based on the current observation \( w_t \). If \( |k_t| \) is small, which will occur if \( |\Sigma_{t|t-1}| \) approaches zero, the new state mean will be mainly estimated based on the state mean prediction \( \mu_{t|t-1} \) [3]. Thus, if we have low observation variance \( r \) we trust the observations, while if we have low state variance predictions \( \Sigma_{t|t-1} \) we trust the model’s predicted state mean. Starting at the update step for \( t = 1 \), state posteriors can be recursively calculated using the prediction step followed by the update step for all \( t \).

For offline settings where all data \{\( w_1, \ldots, w_T \)\} is collected before inference, we use a Kalman Smoother after the Kalman Filter to improve the state estimation. The Kalman Filter only has knowledge of past observations to estimate the distribution at time \( t \) \( Pr(x_t|w_{1:t}; \theta) \), while the smoother knows both past and future observations \( Pr(x_t|w_{1:T}; \theta) \). This leads to an additional step: \( Pr(x_t|w_{1:T}; \theta) = N(\mu_{t|T}, \Sigma_{t|T}) \),

\[
\mu_{t|T} = \mu_{t|t} + \Sigma_{t|t} G_{t+1|t}(\mu_{t+1|T} - \mu_{t+1|t})
\] (4.15)

\[
\Sigma_{t|T} = \Sigma_{t|t} + \Sigma_{t|t} G_{t+1|t}(\Sigma_{t+1|T} - \Sigma_{t+1|t})(\Sigma_{t|t} G_{t+1|t})^T
\] (4.16)

where \( \mu_{t|T} = \mathbb{E}[x_t|w_{1:T}] \) and \( \Sigma_{t|T} = \mathbb{E}[(x_t - \mu_{t|T})(x_t - \mu_{t|T})^T|w_{1:T}] \). Starting with \( t = T \), using the final Kalman Filter estimate of \( \mu_{T|T} \) and \( \Sigma_{T|T} \) from before, we recursively update the smoothed state estimates backwards until \( t = 1 \). In figure 4.4 is a synthetic example of observed data and the inferred distribution using Kalman Smoother.
Figure 4.4: Comparison of the inferred seasonality (posterior distribution) from Kalman Smoother compared to actual seasonality governing connections between nodes of type $x$ and $y$. The underlying seasonality follows a simple sine wave that is accurately recovered from noisy observations.
4.2.3 Maximum Likelihood Estimation

The filtering and smoothing inference algorithms assume knowledge of $Q$ and $r$ from $\theta$ a priori. These parameters are estimated via maximum likelihood estimation (MLE) with a numerical optimization technique. After applying either the Kalman Filter or Kalman Smoother, we have the loglikelihood for a block of the SDSBM. Using this we define an optimization problem to maximize the loglikelihood of the data $w_{1:T}$ with respect the transition variance $Q$ and observation variance $r$:

$$
\arg \max_{Q,r} \log L(w_{1:T}|Q, r) = \arg \max_{Q,r} \sum_{t=1}^{T} \log \mathcal{N}(h_{t}|T, h\Sigma_{t}|T h^T + r)
$$

(4.17)

Starting with initial guesses for $Q$ and $r$, and using one of an array of standard numerical optimization techniques such as Nelder-Mead or BFGS, locally optimal estimates can be achieved. Intuitively, we can imagine unrolling the entire recursive procedure to arrive at a differentiable function of $Q$ and $r$, assuming the data $w_{1:T}$, initial state mean $\mu_{1:0}$, and state variance $\Sigma_{1:0}$ are fixed. The full technical details are available in [6] and [10]. We also explored expectation-maximization updates to the space space model in the next section based in past work [25], but have since found better results by MLE.

4.2.4 Expectation-Maximization

An alternative procedure for estimating time-invariant parameters of $\phi := \{r, Q, \mu_0, \Sigma_0\}$.

The SSM offers a natural form for learning with the expectation-maximization (EM) algorithm to get locally optimal point estimates. EM is suitable for problems where the model depends on the parameters of statistical models, and on latent variables. In the SSM, the $\Theta$ for all $t$ are the statistical parameters, and the $\mu_{t|T}$ and $\Sigma_{t|T}$ are the latent variables (or $\mu_{t|t}$ and $\Sigma_{t|t}$ if using just Kalman Filter). The setup for EM on the standard SSM is discussed in
detail in [2]. We will now define the steps of the EM algorithm, with the addition of the new
variance term defined in equation 4.4. Let us denote the estimated parameters of the $i$th
iteration of EM as $\phi^i$. Initial guess from $\phi^0$ are provided by the user and can be default set
to 1 if no domain context is available. EM works in two steps of expectation step followed
by maximization step, then iterating repeated between the two until convergence.

First define the full log likelihood using the SSM transition and observation equations:

$$
\ln Pr(x_{0:T}, w_{1:T} | \Theta) = \ln \mathcal{N}(\mu_0, \Sigma_0) + \sum_{t=1}^{T} \ln \mathcal{N}(Gx_{t-1}, Q) + \sum_{t=1}^{T} \ln \mathcal{N}(hx_t, r))
$$

(4.18)

The elegance of using EM with SSM becomes clear as the $\phi$ parameters are separated
nicely between the three log likelihood terms. Now in the expectation step, calculate the
expected value of the log likelihood function with respect to the conditional distribution:

$$
Q(\phi, \phi^i) = \mathbb{E}_{x_{0:T} | \phi^i} \left[ \ln p(w_{1:T}, x_{0:T}; \phi) \right]
$$

(4.19)

Using the Kalman Filter and Kalman Smoother equations defined earlier, we recursivly estimate the latent variables. In the end, collect results of:

$$
\mathbb{E}[x_t x_{t-1}^T] = J_{t-1} \Sigma_{t|T} + \mu_{t|T} \mu_{t-1|T}^T
$$

(4.20)

$$
\mathbb{E}[x_t x_t^T] = \Sigma_{t|T} + \mu_{t|T} \mu_{t|T}^T
$$

(4.21)

$$
J_t := \Sigma_{t|T} G^T \Sigma_{t+1|T}^{-1}
$$

(4.22)

This completes the expectation step. Now in the maximization step, find the $\phi$ param-
eters which maximize this log likelihood. More formally:
\[ \phi^{i+1} = \arg \max_{\phi} Q(\phi, \phi^i) \quad (4.23) \]

As the \( \phi \) parameters are well separated in the likelihood, the maximization derivations can be completed separately for each term. For the initial guesses \( \mu_0 \) and \( \Sigma_0 \), the simple updates are:

\[ \mu_0^i = \mu_{0|T} \quad (4.24) \]
\[ \Sigma_0^i = \Sigma_{0|T} \quad (4.25) \]

For the observation variance \( r \), the parameter is set by finding \( r \) which maximizes the function:

\[
\sum_{t=1}^{T} \ln(N(hx_t, r)) = -\sum_{t=1}^{T} \frac{\ln |r|}{2} - \frac{1}{2} \sum_{t=1}^{T} \frac{(w_t w_t^T - w_t \mathbb{E}[x_t] \mathbb{E}[x_t] (h) (h)^T + h \mathbb{E}[x_t x_t^T] (h)^T)}{(r)}
\]

This task can be completed using standard optimization routines such as gradient descent.

For the final variance matrix \( Q \), some modifications need to be made due to not being full rank (only the first two diagonal elements are nonzero). This approach is originally described in [9]. Augment the state vector with two additional elements such that \( x_t^* := [x_t^T, s_{t-d}, m_{t-1}]^T \). To remain valid in the original SSM formulation, modifications also need to be made for \( G, h, \) and \( Q \).
For $h^*$ append two additional zeros, and for $Q^*$, also append additional zeros. This two modification have no effect on the computation, and are just necessary for correct dimensions.

Finally with the the augmented state vector, we can analytically find the $Q^*$ which maximizes $\sum_{t=1}^{T} \ln \mathcal{N}(Gx_t, Q)$, with solutions:

$$q_m = \frac{(d_1 \mu^*_{1|T})^2 + d_1 \sum_{t=1}^{T} \Sigma^*_{1|t} d_1^T}{T}$$  \hspace{1cm} (4.27)

$$q_s = \frac{(d_2 \mu^*_{2|T})^2 + d_2 \sum_{t=1}^{T} \Sigma^*_{2|t} d_2^T}{T}$$  \hspace{1cm} (4.28)

where $d_1 = [1, 0, \ldots, 0, -1]$ and $d_2 = [0, 1, 0, \ldots, 0, -1, 0]$.

Now if we iteratively perform the expectation and maximization steps, we will converge to a locally optimal solution for $\phi$. Given network data $\mathcal{A} = \{A_1, A_2, \ldots, A_t\}$, vertex types, length of seasonality $d$, and initial guesses for $Q$, $r$, $\mu_0$, and $\Sigma_0$, the seasonality of a dynamic network will be extracted.
Results

Evaluation of the SDSBM is completed on both synthetic and real datasets\(^1\). We choose three recent stochastic blockmodels to compare against, namely the Dynamic SBM with known classes (DSBM) [33], its extension for link prediction using an exponentially weighted moving average (DSBMEWMA) [33], and the stochastic block transition model with a similar EWMA extension (SBTMEWMA) [32]\(^2\). Default hyperparameter values found in the code were applied for each model. We baseline the results with a static stochastic block model (SSBM) fit via the maximum likelihood estimator of the block density at time \(t\) as \(p_t^{(a,b)} = \frac{w_t^{(a,b)}}{\mu^{(a,b)}}\). Implementation of the SDSBM uses the *Kalman Filtering and Smoothing* package in *R* [10]. The length of seasonality is set to \(d = 3\) for the synthetic experiments and \(d = 7\) for the real datasets. The remaining SDSBM hyperparameters are assigned values that do not impact fitting, specifically \(\mu_{1.0}^{(a,b)} = 0\), \(\Sigma_{1.0}^{(a,b)} = \text{diag}[1000]\), initial \(Q^{(a,b)} = \text{diag}[1]\) and initial \(r^{(a,b)} = 1\).

\(^1\)Reproducible evaluation code will be publicly available after review process

\(^2\)Implementations of DSBM, DSBMEWMA, and SBTMEWMA are provided by author Professor Xu at bit.ly/2qK0tp
5.1 Synthetic Experiments

We evaluate the potential of the SDSBM, DSBMs, and SSBM to recover the true time-varying SBM block density parameters from simulated networks. The SBTMEWMA has additional parameters over the SBM that prevent it from being used in this comparison. We define a matrix $V_t$ of true SBM block density parameters, where each element $[V_t]_{ab} = v_t^{(a,b)}$ defines an expected edge density between nodes of type $a$ and $b$. We generate directed dynamic networks of $n = 128$ vertices, with $k = 4$ fixed classes of 32 vertices each, and $T = 30$ time steps. $A$ is generated following Equation 4.5, where $p_t^{(a,b)} = v_t^{(a,b)} + \epsilon_p^{(a,b)}$, and $\epsilon_p^{(a,b)} \sim \mathcal{N}(0, \sigma^2)$. Two settings are considered:

1. In the parametric noise experiment, we fix the seasonal pattern to $v_t^{(a,b)} = c^{(a,b)}$, where $c^{(a,b)} \sim \text{Uniform}(0.05, 0.95)$, and vary the amount of noise $\sigma$ within $[0, 0.15]$.

2. In the parametric seasonality experiment, we fix noise to $\sigma = 0$ and vary the amplitude of a seasonal pattern from $v_t^{(a,b)} = \alpha \sin(\frac{2\pi t}{T})$, with $\alpha$ between $[0, 0.45]$.

The former setting tests the ability of the models to recover the true parameters under non-seasonal, noisy data. The latter setting tests the models’ ability to recover true parameters under seasonal patterns of varying amplitude. Because the DSBM does not perform well for $p_1$ near 0 or 1 we do not consider these edge cases (SDSBM does not have the same issues near 0 and 1). To evaluate accuracy of each model, we use mean squared error (MSE) between the true SBM parameters $V_t$ and estimated SBM parameters $\hat{V}_t$:

$$\frac{1}{T} \sum_{t=1}^{T} ||V_t - \hat{V}_t||_2^2$$

across all time steps. The estimated $\hat{V}_t$ for the SDSBM is $h\mu_{t|T}$, for the DSBM is $h(\psi^t)$ (where $h()$ and $\psi^t$ are as defined in [33]), and SSBM is the maximum likelihood estimate. The SDSBM is fit by Kalman Smoothing as the dynamic network for all $t$ is observed.

The SSBM is an effective baseline to determine the usefulness of including modeling of temporal information. In Figure 5.1(a), we can see the inclusion of this information...
is helpful for both the DSBM and SDSBM at all noise levels. As the synthetic data becomes noisier, the temporal information becomes more valuable to accurately model the data. The reason the SDSBM performs better than the DSBM is likely attributed to the Kalman Smoother, as it incorporates additional information from both past and future data at each time step. The results from Figure 5.1(b) illustrate the benefit of using the SDSBM to model seasonal changes in network dynamics. When the dataset exhibits even a slight amount of seasonality of amplitude 0.05, the existing DSBM already performs worse than the baseline SSBM. The SDSBM is able to recover the simulated SBM parameters on data of any amplitude. These experiments provide the insight that the SDSBM will be most effective over existing models on real datasets with seasonal patterns with larger amplitudes. In the next section, this statement is supported by accurate edge counts prediction results on several real datasets.
5.2 Evaluation on Real Data

In dynamic networks, a common task is to predict if edges will exist at time $t + 1$ given observations up to time $t$. In the following experiments, we evaluate the model’s capability to predict future edge counts (not to be confused with individual edge prediction as more commonly tested [32, 33]). Any variant of the SBM assumes edges of the same vertex types are independently and identically distributed, and is therefore only suited for block level predictions. We also apply the SDSBM to dynamic network anomaly detection over New York City (NYC) Taxi trips and E-mail communications.

The datasets used in the edge counts prediction experiments include months of Taxi trips across NYC\(^3\) (nodes represent city districts, node types are the five Boroughs of NYC \{Bronx, Brooklyn, Manhattan, Queens, Staten Island\}, and an edge indicates a trip from one district to another), U.S. Airline\(^4\) (nodes represent U.S. airports, node types are three airport size classifications \{small, medium, large\} and an edge indicates a flight leg between two airports), and the classic Enron e-mail network [15] (nodes represent people, node types are seven job titles \{Director, CEO, President, Vice President, Manager, Trader, Other\} (which merge employees, in house lawyer, and nodes without labels)), and an edge if email from one user to another, in either to, cc, or bcc.). Summary information about the dynamic networks in these datasets are in Table 5.1. All datasets are discretized daily. Length specifies the total number of discrete time periods we separated the data into, thus each dataset is represented by a dynamic network $\mathcal{A} = \{A_1, A_2, \ldots, A_T\}$. $\mathbb{E}[edges/day]$ specifies the average number of edges observed in each $A_i \in \mathcal{A}$. All networks are directed, so if there are $k$ node types, there are $k^2$ blocks.

\(^3\)NYC Taxi dataset available: on.nyc.gov/1EjFCfd
\(^4\)Airline dataset available: bit.ly/2HbO2Sx
Table 5.1: Overview of real datasets

<table>
<thead>
<tr>
<th>Name</th>
<th>Nodes</th>
<th>$E[\text{edges/day}]$</th>
<th>Types</th>
<th>Blocks</th>
<th>Length $T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYC Taxis</td>
<td>262</td>
<td>4848</td>
<td>5</td>
<td>25</td>
<td>181</td>
</tr>
<tr>
<td>US Airline</td>
<td>291</td>
<td>3774</td>
<td>3</td>
<td>9</td>
<td>364</td>
</tr>
<tr>
<td>Enron</td>
<td>184</td>
<td>15</td>
<td>7</td>
<td>49</td>
<td>1633</td>
</tr>
</tbody>
</table>

Figure 5.2: Seasonal time series of edge counts in each dataset.

### 5.2.1 Edge Count Prediction

We evaluate the SDSBM, DSBM, DSBMEWMA, and SBTMEWMA on their ability to predict edge counts. As the DSBMEWMA and SBTMEWMA predict the probability of individual edges, we define a binary threshold at 0.5, converting the predictions to a binary representation that allows for calculation of edge counts. Each of the datasets naturally form a seasonal pattern week to week as qualitatively seen in Figure 5.2. The SDSBM is trained using the Kalman Filter as we are predicting on streaming data.

To evaluate the edge prediction we use mean absolute error (MAE) defined as:

$$ MAE = \frac{\sum_{t=s}^{T} |y_t - \hat{y}_t|}{T-s+1} $$

where $y_t$ is the number of observed edges in the full network at time $t$, $\hat{y}_t$ is the predicted number of edges at time $t$ given the model trained to time $t - 1$, $|y_t - \hat{y}_t|$ is the absolute error (AE) at time $t$, and $s$ is the starting time step for calculation. We wait for one full seasonal period before calculating MAE as the SDSBM requires at least one observation period to form an initial seasonal pattern.

Figures 5.3(a), 5.3(b), and 5.3(c) shows the MAE results for each dataset and SBM model. In understanding the difference in performance between datasets, it is helpful to review their seasonal patterns in Figure 5.2. The NYC Taxi data has the largest amplitude.
in its seasonal pattern, with most of the days in the weekly pattern having different edge count values. This is opposed to the Airline and Enron datasets, where six and five of the days in each season have similar values, with the remaining days having distinct lower values. From the earlier analysis of the synthetic results, the relative performance of the SDSBM is expected to be the best on the NYC data, as it has the seasonal pattern of largest amplitude. This expectation is matched in Figure 5.3(a) where the SDSBM is over twice as accurate as the next best model on NYC taxi data. We have previously shown the DSBM does not perform well on seasonal datasets, and this issue repeats with the DSBMEWMA and SBTMEWMA. The EWMA extensions are intended to provide a smooth mean fit to the data, but doing this ignores seasonal structure, missing information to improve predictions. The SBTMEWMA is built on the intuition that whether an individual edge exists at the previous time step may be indicative to the existence of the same edges at the next time. The poor performance suggests this assumption may not be valid in seasonal datasets.

We further examine the absolute error at each time step for the NYC dataset in Figure 5.3(d). Errors that are dependent or correlated with each other violates their i.i.d. assumption the model makes. To check for this violation, we quantitatively test for autocorrelation in the error. In Figure 5.3(e), the autocorrelation shows significant dependency among for the DSBM and DSBMEWMA at time lags in multiples of 7, aligning with the seasonal pattern, while the SDSBM has minimal autocorrelation. This correlation is also clearly visible when examining the errors from the two models. The inability of the DSBM and DSBMEWMA to model the seasonal structure is thus leading to the worse estimates.

5.3 Anomaly Detection Use Case

We illustrate a practical application of the SDSBM for identifying anomalies in the dynamics of networked systems. Once fitted to a history of past observations of a system, the model can evaluate its loglikelihood against true observations at a particular time to mea-
Figure 5.3: In panels (a), (b), and (c) are the edge counts prediction results for each model on the three datasets. In panel (d), the absolute error (AE) on NYC Taxis across time is presented, providing qualitative evidence for unmodeled structure in the DSBM and DSBMEWMA. In panel (e), the autocorrelations (AC) for the absolute errors on NYC Taxis with 95% confidence intervals (CI) are calculated, providing further quantitative evidence for unmodeled structure in the DSBM and DSBMEWMA.
sure the extent to which the system is acting anomalous. We use the SDSBM to report the $c$ most anomalous time steps (e.g. the time steps with lowest loglikelihood) with $c$ decided by the user. As the Kalman Smoother formulation defines a loglikelihood per block, from Equation 4.17, we define the loglikelihood for a single time step $t$ of an observed dynamic network $A$ by:

$$\log \mathcal{L}(A_t) = \sum_{\forall ab} \ln Pr(w_{1:T}^{(a,b)}; \theta^{(a,b)}) = \sum_{\forall ab} \ln \mathcal{N}(\mu_{t|T}^{(a,b)}, \Sigma_{t|T}^{(a,b)} h^T + r^{(a,b)})$$

(5.1)

log $\mathcal{L}(A_t)$ allows the user to evaluate both the loglikelihood of a network at time $t$, and the loglikelihood contribution for each individual block $(a, b)$. The combination of both network level (the results of the full summation) and block level (the summand of a specific block $(a, b)$) information helps a user narrow the source and location of an anomaly. We establish a framework to use the SDSBM to detect and explore anomalies:

1. Given observed dynamic networks $A$ with known class labels, fit the SDSBM from Chapter 4.

2. Use Equation 5.1 to calculate the network level and block level loglikelihoods at each time step.

3. For the $c$ time steps with lowest $\log \mathcal{L}(A_t)$:

   (a) Explore block level loglikelihood values for patterns, e.g. identifying if all the blocks have similar loglikelihoods, or if some blocks have much lower values. The former case suggests a network-wide anomaly, while the latter case is localized.

   (b) Qualitatively explore the observed and SDSBM fit to a time series of edge counts for several of the blocks, looking for patterns like large deviations in
observed edge counts from the model’s seasonal fit.

(c) Find explanations for the anomaly from external sources, taking advantage of the known time of the anomaly patterns within the block level loglikelihoods and edge count time series.

We demonstrate this framework with examples over NYC Taxi trips and the Enron E-mail network.

We developed a software tool GEONET, to allow for easy visualization and exploration of the anomalous time steps following the above algorithm. Along with the implementation of the SDSBM anomaly detection, the user is able to explore a network placed on a leaflet geospatial map, and visual that network across time with interactive sliders. An image of the tool being applied to the New York City data is shown in figure 5.4.
5.3.1 NYC Taxi Trips

We detect and explore several abnormal days over a six month period of NYC taxi trips in 2017. Recall that edges in the dynamic network representation of taxi trips represent a trip from one city district to another, and that districts are blocked by the five boroughs of the city. Figure 5.5 shows network and block level loglikelihoods to an SDSBM fitted to a dynamic network representing daily taxi trips across the city. The days with lowest loglikelihood represent intuitively anomalous days such as New Years, numerous snow storms, long taxi trips to Staten Island, and notable New York Yankees baseball games.

We first explore the New Year’s and two snow storm anomalies. Figure 5.5 find all blocks to have similar loglikelihood, indicating some sort of network wide event. In Figure 5.6, we see a time series of the number of edges in Queens to Queens block. There are
three distinct large deviations of the observed from the fitted seasonal pattern, with higher than expected taxi trips on New Year’s, and drastically lower than expected taxi trips during the February 9th and March 14th snow storms.

We also investigate the May 14th anomaly. The loglikelihood plot in Figure 5.5 show that the Bronx-Manhattan and Bronx-Queens blocks are the ones with lowest low loglikelihood. Examining the network on this day overlaid on a visual map of New York City in Figure 5.7 finds a single node with unusually high degree that is incident to a number of Queens and Manhattan nodes, located at the New York Yankees baseball stadium. A news check explains the source of this anomaly: legendary player Derek Jeter’s baseball number retirement ceremony was being held at the stadium that day.\footnote{An article on the game atmlb.com/2qJe2JG}

## 5.3.2 Enron E-mails

For the anomaly detection over the Enron E-mails, we reduce the number of blocks from 49 down to 9, improving the quality of SDSBM fits by reducing the data sparsity of each
Figure 5.7: A node with anomalous degree in the Bronx on May 14th, 2017.

Figure 5.8: Time series of edge counts between blocks of the Enron e-mail network with anomalies.
Table 5.2: Loglikelihood of 4 most anomalous time steps, with the three most anomalous blocks per time step shown in bold.

<table>
<thead>
<tr>
<th>Time</th>
<th>low-low</th>
<th>low-mid</th>
<th>low-high</th>
<th>mid-low</th>
<th>mid-mid</th>
</tr>
</thead>
<tbody>
<tr>
<td>05-22-2001</td>
<td>-107.92</td>
<td>-2.32</td>
<td>-21.32</td>
<td>-5.45</td>
<td>-63.3</td>
</tr>
<tr>
<td>08-23-2001</td>
<td>-9.95</td>
<td>-2.09</td>
<td>-19.74</td>
<td>-73.89</td>
<td>-42.92</td>
</tr>
<tr>
<td>09-13-2001</td>
<td>0.30</td>
<td>-15.06</td>
<td>-1.39</td>
<td>-1357.87</td>
<td>-622.12</td>
</tr>
<tr>
<td>10-05-2001</td>
<td><strong>-629.10</strong></td>
<td><strong>-2251.92</strong></td>
<td><strong>-770.54</strong></td>
<td>-23.49</td>
<td>-53.50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>block</th>
<th>mid-high</th>
<th>high-low</th>
<th>high-mid</th>
<th>high-high</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.17</td>
<td>-1543.28</td>
<td>-2520.44</td>
<td>-441.11</td>
<td></td>
</tr>
<tr>
<td>-1.78</td>
<td><strong>-562.02</strong></td>
<td>-1111.76</td>
<td>-206.84</td>
<td></td>
</tr>
<tr>
<td><strong>-140.89</strong></td>
<td>-10.038</td>
<td>-66.63</td>
<td>-5.32</td>
<td></td>
</tr>
<tr>
<td>-33.54</td>
<td>0.080</td>
<td>-41.90</td>
<td>-9.47</td>
<td></td>
</tr>
</tbody>
</table>

block. This was completed by merging the original job titles to three groups based on low {Employee, Trader}, mid {Manager}, and high {CEO, Director, Managing Director, President, Vice President} ranking job titles (the single in house lawyer and users with missing job titles were dropped from the analysis).

The loglikelihoods per block for the four most anomalous time steps are show in in Table 5.2. The block-level information provides valuable insight that each of the anomalies were localized to a single node type, as shown by the grouping of low loglikelihoods emphasized in bold (e.g. the May 22nd anomaly has significantly lower loglikelihoods for the blocks with ‘from’ node type ‘high’). Following the anomaly exploration framework defined earlier, the next step is to explore the time series of edge counts for some of the anomalous blocks. Figures 5.8(a), 5.8(b), and 5.8(c) confirm the localization of the anomalies, with the first figure having largest spikes on May 22nd and August 23rd, the second with a spike on September 13th, and the last having a spike on October 5th.

We explain the anomaly sources by referring to outside sources. According to Koutra et al., the May 22nd anomaly aligns with high level person Jordan Mintz sending memorandum to another high level person Jeffrey Skilling for his sign-off on the LJM paperwork, the August 23rd anomaly corresponds to upper management looking to focus the company to restore investor confidence, and the October 5th anomaly revealing chatter between low
employees and their managers immediately before law firm Davis Polk & Wardwell is hired to prepare legal defense of the company [16].

5.4 Theoretical Analysis

Synthetic experiments are completed to explore the impact of adding the additional measurement noise parameter from Equation 4.4. The state space model as defined in Equations 4.7 and 4.6 has two variance parameters, $Q$ and $r$. $Q$ contains $\{q_m, q_s\}$ which together define the level of transition noise which controls the noise in the latent seasonality. $r$ in our definition is defined as the sum of $u$ the observation noise inherently created and controlled by the probability of the binary random decision process, and $b$ the level of measurement noise in the probability of the decision process. Existing applications of the state space model such as in [33] do not have the measurement noise (so $b = 0$).

To understand how this parameter will affect forecasts in the SSM, we setup a simple synthetic experiment. Data is generated using the state space model formulation with a small $Q$, and the level of noise to be expected in real data (a ‘medium’ amount) for $r$. First in figure 5.9, the SDSBM is fit using the Kalman Filter and EM with $b$ being fixed to zero. This original form of the model will have to place the synthetic measurement noise into the optimization of $Q$. This will results in a larger $Q$ than is reasonable. When forecasting into the future, the large $Q$ results is rapidly large confidence bounds. Given that the original data never became close to value 3000, it is not useful to have confidence bounds that wide. The forecasting is essentially useless.

Now if we complete the same fitting, this time with a nonzero $b$, as shown in figure 5.10. The model is able to keep the $Q$ variance low, while capturing the additional variation in $r$. This effect is clearly seen in the confidence bounds of the forecast. The smaller bounds result in much more useful forecasting.
Figure 5.9: This is a plot of data generated and fit from the typical state space model with only the transition noise and observation noise, without the additional measurement noise parameter. The time series of edge counts data is in black, forecasted mean fit in red, and 95% confidence bounds in green.
Figure 5.10: This is a plot of data generated and fit from the new state space model with all three of the noise terms. The time series of edge counts data is in black, forecasted mean fit in red, and 95% confidence bounds in green.
The main difference in $Q$ and $r$ in forecasting, is the uncertainty due to $Q$ cumulates with time, while uncertainty due to $r$ only occurs at the specific time step $t$. If a forecasting is predicting $y$ time steps into the future, the confidence bounds will be proportional to $y \times Q + r$.

In order for these result to be useful, we need to argue the noise from $r$ will exist and be significant in real dataset. The original problem this model has been developed for is in a surveillance application. Data of movement in a city is collected from a single wide area sensor. In this complex system, we can expect process noise of the seasonal process as people follow the daily patterns, there is measurement noise from the sensor itself, and there is the observation noise from the uncertainty in individuals’ movements. Without the additional parameter, the uncertainty due to the sensor should result in under-confident forecasts.
6

Conclusion

As more dynamic network datasets and application areas become available, better techniques for modeling must be developed. The current popular modeling techniques in dynamic networks rely on some network feature such as the number of edges to follow a random walk through time. When looking to use dynamic network modeling for movement in large geospaces, the random walk approaches existing were insufficient. These systems are driven by structured temporal patterns, with predictable rises (e.g. rush hour) and falls (late night) in movement. By using seasonal techniques from time series, we were created the seasonal dynamic stochastic blockmodel to more effectively model these datasets..

This new statistical model for dynamic networks was created by fusing the benefits of seasonal structural time series with stochastic blockmodels. The generative specification and inference procedure are defined and utilized. The improved capabilities of the model on seasonal datasets over existing models is demonstrated on several synthetic and applied datasets. For edge density prediction, the model is up to twice as accurate as competing models on the New York City taxis dataset, with supporting arguments showing this severity of improvement is dependent on the amount of seasonality in the data. Additionally the SDSBM provides a new way to perform anomaly detection on dynamic networks. The
model is able to capture an intuitive notion of the ‘pattern of life’ of a geospatial systems, and define deviations from this pattern as anomalous. The generality of the model was also demonstrated with an application of the anomaly detection on the Enron email network.

**Future Work**

There are several methodological, computational and applied directions this work can be extended. In the research community, two of the most frequently seen applications of the stochastic blockmodel is for individual edge prediction (link prediction) and inferring the vertex labels (community detection). In this thesis, the original motivation for the model was to capture seasonality to detect anomalies. The model is able to predict edge density, but is currently not able to predict individual edges. To bring the work in line with the typical approaches, it is possible to extend the SDSBM to be applicable to both problems. One simple approach to allowing link prediction is to follow [33], by creating a weighted combination of the block level information with a time series fit on each individual edge. Although this approach would allow link prediction, it comes with several problems. In the evaluation chapter 5, we showed the direct application of the EWMA was not effective for the seasonal datasets. Additionally, the approach does not fit into the statistical and generative nature of the model, and makes evaluation of the likelihood of the fit more difficult. Researching a generative approach that still allows link prediction would be an interesting question. To modify the SDSBM for community detection, we can again look to [33]. For both the SDSBM and the DSBM, given vertex labels and data, an evaluation of the full likelihood of the model is possible. By creating a simple hill climbing algorithm, where you change one vertex label, evaluate the likelihood, and keep the change if it improves the likelihood, an algorithm for inferring vertex labels is created.

Although the SDSBM has many user settable hyperparameters and parameters, the only parameter shown to have significant impact on the modeling performance is the length of seasonality. This parameter must be exactly correct, otherwise the seasonal fit will not
be accurate. In all applications explored so far, the data had natural seasonality that is determined from domain specific knowledge. For datasets where seasonality is not easily determined, there have been examples of determining the seasonality using a periodogram. The periodogram calculates the spectral density of a signal. Thresholding to detect for prominent spikes in the periodogram could define a heuristic to detect seasonality.

To model the seasonality in this work, we used the technique of discrete dummy seasonal offsets. In [6], the authors also present another method of capturing seasonality using a Fourier series and claim it is preferable in practice.

Vertices were assumed to be fixed through time in the model. In many dynamic networks, nodes will enter and leave the system over time. By simply allowing the block sizes \( n \) to vary with time, the same SDSBM algorithm can be used in new domains. The effect of this change on modeling accuracy would need to be explored and understood.

In static network research, significant effort has been research applied to ensure distributions of network properties of models accurately reflect properties found in real data. With ideas like power-law distributions, scale-free networks, and small world networks, the importance of these distributions have been established. It is known the SDSBM will generate networks that do not accurately reflect the degree distributions found in real data due to the simplistic IID assumptions. Degree corrected dynamic variants of the SBM exist [28] and those ideas could be brought into the SDSBM.

A current computational issue with the SDSBM is its inability to scale for very long seasons. In the current implementation, the size of the Kalman Filter matrices grow quadratically with the length of the seasonality, and hence number of operations for matrix multiplication grows cubically. To allow for truly finessed grained modeling, such as minutes or seconds datasets like New York City taxis, a new approach needs to be explored.

An applied concern of many network scientists is whether a model can scale to tens, hundreds, thousands, millions, or even billions of nodes. The SDSBM has only been explored on datasets up to hundreds of nodes. In the New York City dataset, for data before
the year 2017, the pick up and drop off locations were provided by GPS coordinates instead of predefined locations. With the GPS representation in combination with New York census information, other researchers have created dynamic networks with over 38k nodes [22]. It could be fruitful to explore both the edge density prediction and the anomaly detection capabilities at differing number of nodes. With such fine grained nodes, different types of anomalies could be discoverable.
Bibliography


[17] Xiaolei Li, Zhenhui Li, Jiawei Han, and Jae-Gil Lee. Temporal outlier detection in vehicle traffic data. In *Intl. Conference on Data Engineering*, pages 1319–1322. IEEE, 2009.


