Holistic Learning for Multi-Target and Network Monitoring Problems

by

Bahareh Azarnoush

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2014 by the
Graduate Supervisory Committee:

George C. Runger, Co- Chair
Jennifer Bekki, Co-Chair
Rong Pan
Soroush Saghafian

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

Technological advances have enabled the generation and collection of various data from complex systems, thus, creating ample opportunity to integrate knowledge in many decision-making applications. This dissertation introduces holistic learning as the integration of a comprehensive set of relationships that are used towards the learning objective. The holistic view of the problem allows for richer learning from data and, thereby, improves decision making.

The first topic of this dissertation is the prediction of several target attributes using a common set of predictor attributes. In a holistic learning approach the relationships between target attributes are embedded into the learning algorithm created in this dissertation. Specifically, a novel tree-based ensemble that leverages the relationships between target attributes towards constructing a diverse, yet strong, ensemble is proposed. The method is justified through its connection to existing methods and experimental evaluations on synthetic and real data.

The second topic pertains to monitoring complex systems that are modeled as networks. Such systems present a rich set of attributes and relationships for which holistic learning is important. In social networks, for example, in addition to friendship ties, various attributes concerning the users gender, age, topic of messages, time of messages, etc. are collected. A restricted form of monitoring fails to take the relationships of multiple attributes into account, whereas the holistic view embeds such relationships in the monitoring methods. The focus is on the difficult task to detect a change in only a subregion of a high-dimensional space of network attributes that requires an integrated, holistic learning approach. One contribution is a monitoring algorithm based on a network statistical model that is elaborated on synthetic and real networks. Also, a generalizable model to monitor an attributed network is presented that transforms the task into an expedient structure for a machine learning

algorithm. A learning step in this algorithm manages changes that may only be local to subregions (with a broader potential for other learning tasks). The model and algorithm are integrated to contribute a holistic, robust, generalizable monitoring method. Evaluations on synthetic and real networks are provided.

DEDICATION

To my family

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor Professor George Runger for his guidance, support and encouragement throughout my Ph.D. studies. I truly admire his intellectual thinking and strong work ethic and am extremely grateful for everything I have learned from him.

I would also like to thank my committee members Professor Jennifer Bekki, Professor Rong Pan and Professor Soroush Saghafian for their valuable contributions to my dissertation.

Last but definitely not least, I would like to thank my family, without whom none of this would be possible. Words cannot express my gratitude for their unconditional love, support and guidance throughout all the stages of my life. I am forever indebted to them.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Technological advances have enabled the generation and collection of various data from complex systems. Examples include daily data collected from social networks, manufacturing, educational and financial systems. In social networks, for example, in addition to friendship ties, various data concerning the users' gender, age, topic of messages, time of messages, etc is collected. Effective learning from data allows for the integration of knowledge in myriads of decision-making applications. This dissertation introduces holistic learning as the integration of a comprehensive set of relationships that are used towards the learning objective. The holistic view of the problem allows for richer learning from data and, thereby, improves decision making.

The first topic of this dissertation is the prediction of several target attributes using a common set of predictor attributes. Problems of this type arise naturally in different fields such as manufacturing: predicting various quality aspects of a product using the manufacturing settings (Breiman and Friedman, 2002), marketing: predicting different aspects of consumer behavior based on consumer characteristics (Zhang *et al.*, 2005) and education: predicting different learning outcomes based on learning activity (Azarnoush *et al.*, 2013; Tatsuoka and Lohnes, 1988).

Such problems are usually associated with two types of training data. In the first type, the data set has M predictor attributes, T target attributes, and N instances in the form (x_i, y_i) for $i = 1, \cdots, N$, $x_i = (x_{i1}, \ldots, x_{iM})'$ and $y_i = (y_{i1}, \ldots, y_{iT})'$ which is usually referred to as multi-target learning (Blockeel *et al.*, 1998). The second type is T separate data sets in the form of (x_i, y_i) where $x_i = (x_{i1}, \ldots, x_{iJ})$ and y_i is a single numerical value which is usually referred to as multi-task learning (Caruana, 1998).

Tree models are widely used learning algorithms that recursively partition the instances at the nodes into homogeneous child nodes. As an example, classification and regression trees (CART) (Breiman *et al.*, 1984) select partitions to minimize the Gini index for classification problems and to minimize the sum of squares for regression problems. The homogeneity is evaluated with respect to a single target attribute. Current methods have extended tree models for multi-target problems by measuring homogeneity with respect to all of the target attributes. For example, Caruana (1993) selected partitions to minimize the average entropy over the target attributes. Similarly, Blockeel *et al.* (1998) selected partitions to minimize the sum of entropies (classification) or the sum of variances (regression) of the individual target attributes. Additionally, De'Ath (2002) selected partitions to minimize the total sum of squares of the target attributes.

A shortcoming of current approaches is the possible disagreement across the target attributes in selecting the optimal partitions. As the number of target attributes increases, fewer partitions in a tree will be optimal for any one target attribute. Methods based on an average of the attributes might not sufficiently consider the relationships between the attributes. In a holistic learning approach, however, the relationships between target attributes are embedded into the learning algorithm. In this direction, we present a new tree-based model that leverages the relationships across multiple target attributes called the compound forest (CF).

The CF method leverages the relationships towards constructing a diverse, yet strong, ensemble by training trees on one target attribute and using it to generate predictions for another. The base learners within this ensemble are grouped based on the target attribute that was used in their training. A sparse group regression model that takes this grouping into account is adopted to assign weights to each base learner. This provides robustness to non-relevant learners between and within

groups. Experiments on synthetic and real data compare CF to related methods and highlight its benefits. We conclude that CF improves prediction performance by leveraging useful relationships across target attributes while remaining robust in the absence of useful relationships.

Network modeling and analysis has become a fundamental tool for studying various complex systems such as social, cyber and biological systems. The second topic of this dissertation pertains to these systems as they present a rich set of attributes and relationships for which holistic learning is important. Specifically, we focus on network monitoring which is usually tailored around two objectives that we refer to as testing for *static homogeneity* and testing for *temporal homogeneity*. Testing for static homogeneity aims to detect networks that have anomalies with respect to the current network (see for example Miller *et al.* (2013)). Testing for temporal homogeneity, on the other hand, aims to detect networks that have anomalies with respect to past networks. This is an important problem as changes in the system are reflected in the network and temporal homogeneity is the focus of the research here.

A typical approach towards testing temporal homogeneity is to monitor extracted measures from the network topology. The topology is the network structure that is induced from the vertices and connecting edges. As an example, McCulloh and Carley (2011) constructed control charts over different network measures such as density, average degree, average closeness and average betweenness. The work by Priebe *et al.* (2005), Marchette (2012) and Neil *et al.* (2014) monitored scan statistics for this purpose. Similarly, the work by Park *et al.* (2013) used a fusion of network statistics (including the scan static) to detect changes in a stream of networks. The cited work are all based on monitoring extracted measures from the network topology.

In addition to the network topology, many real systems present additional layers of data generated through vertex and edge attributes. In an email network, for ex-

ample, the attributes include the role of the sender and receiver, the topic of email, size of email, etc. Monitoring a stream of such networks calls for a method to detect change in any subregion defined by the attributes. An important issue here is the high dimensionality that arises from networks having a large number of attributes. Simultaneous monitoring of the subregions is defeated by the combinatorial explosion of the number of region subsets making this problem especially challenging. A restricted form of monitoring fails to take the relationships of multiple attributes into account, whereas the holistic view embeds such relationships in monitoring methods. The focus is on the difficult task to detect a change in only a subregion of a high-dimensional space of network attributes that requires an integrated, holistic learning approach.

We motivate the problem through the following simple example. Consider the network in Part (a) of Figure 1.1 where edges are homogeneously present on the entire network. An external event results in excessive communication over a small subset of the vertices shown in Part (b) of Figure 1.1. An approach for the detection of such a temporal inhomogeneity is to monitor the network topology over partitions of the network (as done in Priebe *et al.* (2005), Marchette (2012), Neil *et al.* (2014) and Park *et al.* (2013)). This is, however, challenging given the absence of prior knowledge about the location of inhomogeneity.

Besides the topology, many networks include vertex attributes that may be useful for the identification of the change. For example, a social network is composed of friendship ties as well as some attributes such as gender, age, etc. The networks in Figure 1.1 Parts (a) and (b) are revisited in Figure 1.2 by incorporating such attributes (each vertex is associated with a unique ID and two attributes are shown in color and size). These figures shed light on the location of inhomogeneity through the attributes: namely, that the excessive communication is amongst vertices of the

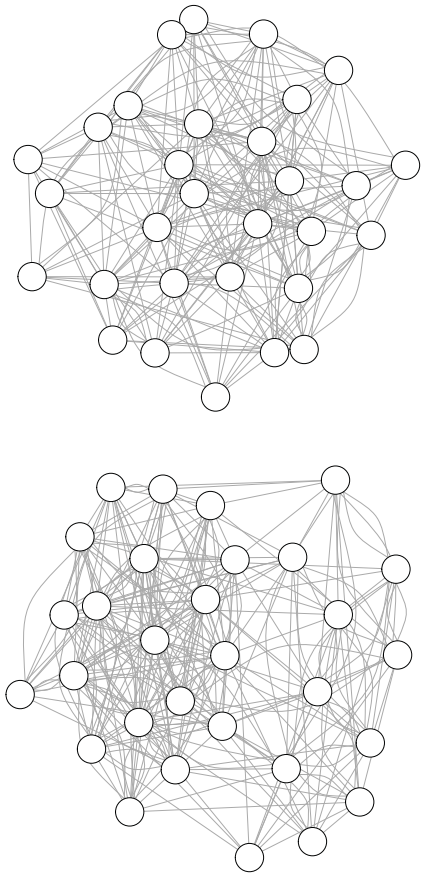


Figure 1.1: An Example Of Change In Networks: Part (a) Network Where Edges Are Homogeneously Present On The Entire Network. Part (b) Change of Local Inhomogeneity.

same color. Note that this type of change is more precisely described as excessive activity in local regions of the attribute space and is, thus, better detected through a holistic monitoring approach that leverages the attribute relationships.

Our work leverages the network attributes and relationships towards extending statistical monitoring to network streams. Chapter 4 presents a method that leverages vertex attributes in modeling and monitoring networks through a logistic regression framework. To this end, a model for the probability of edge existence as a function of vertex attributes is constructed and a likelihood method is developed to monitor the underlying network model.

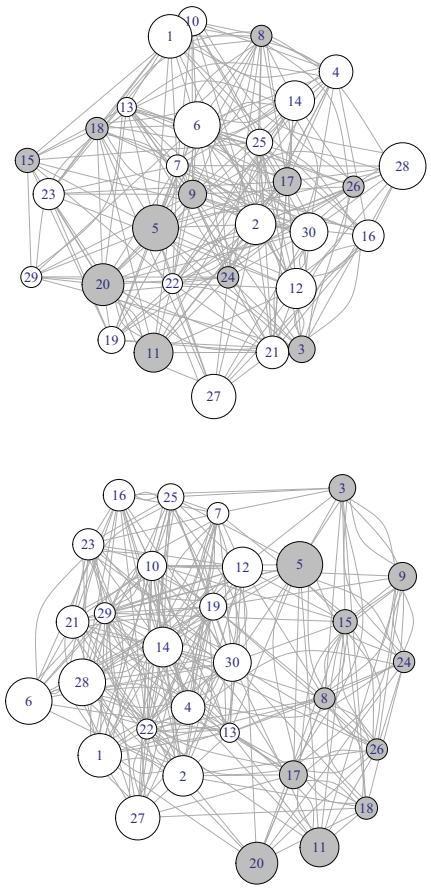


Figure 1.2: The Networks Are Augmented With Attributes Allowing Insight on The Location of Inhomogeneity Through The Attributes.

Chapter 5 continues this topic by introducing a novel method for monitoring networks with various attributes. Vertex, edge and topological attributes are considered. The presented method is based on transforming the network monitoring problem to one of supervised learning. This transformation provides a set of powerful tools that are used towards devising a monitoring approach that effectively detects change in any subregion defined by the attributes that affects only a small subset of the network. Moreover, diagnostic tools that provide insight on the nature of change are derived. Experiments on simulated and real network streams depict the properties and benefits of the methods.

This dissertation is arranged as follows. The next chapter provides a background

on some methods that are utilized in our research and is followed by detailed description of our work in Chapters 3, 4 and 5. Finally Chapter 6 provides concluding remarks and directions for future work.

Chapter 2

BACKGROUND

2.1 Tree-Based Methods

Tree-based methods partition the attribute space into homogeneous regions and fit a simple model to each region. Different tree-building procedures have been proposed with various applications (Rokach, 2008). Trees are used for both regression and classification problems. In regression problems, the goal is to predict a quantitative target, whereas, in classification problems the goal is to predict a qualitative target. Classification and regression tree (CART) (Breiman *et al.*, 1984) is a popular tree-based method that is based on binary recursive splits. At each node, all attribute, value pairs are evaluated for splitting and the one that results in the most homogeneous child nodes is selected. The splitting criterion is the sum of squares for regression and Gini index, misclassification error or cross-entropy for classification (Breiman *et al.*, 1984).

For a regression problem with a data set with M attributes and a quantitative target for N instances: (x_i, y_i) for $i = 1, \cdots, N$, $x_i = (x_{i1}, x_{i2}, \cdots, x_{iM})'$, for example, we select splitting variable j and split point s to solve

$$\underset{j,s}{min}[\underset{c_1}{min} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \underset{c_2}{min} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2] \tag{2.1}$$

Here, $R_1(j, s) = \{X \mid X_j \leq s\}$ and $R_2(j, s) = \{X \mid X_j > s\}$ are half-planes that result from splitting on attribute j at split point s (Friedman *et al.*, 2001).

Tree-based methods have many attractive properties: capture nonlinear relationships, handle missing values, invariance to attribute units and robustness to outliers.

However, one of their major drawbacks is their instability and high variance which has motivated the construction of tree ensembles such as random forest (RF) (Breiman, 2001).

A RF constructs a parallel ensemble of de-correlated trees. Each tree is constructed on a random sample with replacement from the data (bootstrap sample). At each node of each tree, a subset of m candidate attributes from the set of M input attributes is selected for evaluation and the attribute that results in the most homogeneous child nodes is selected. Recommendations are $m = \sqrt{M}$ for classification and $m = M/3$ for regression problems (Friedman *et al.*, 2001). A collection of trees grown in this fashion form a diverse ensemble that results in variance reduction and higher stability.

In addition to the attractive properties of a single trees, RFs offer additional benefits. They have high accuracy and provide estimates of variable importance, generalization error, class-probability estimates.

The RF's variable importance measure is based on a tree's intrinsic feature selection capability. The variable importance of a single tree T is

$$VI(X_j, T) = \sum_{\nu \in T}^{\cdot} \Delta I(X_j, \nu) \tag{2.2}$$

where $\Delta I(X_j, \nu)$ denotes the information gain (Breiman *et al.*, 1984) due to a split on X_j at node ν. For an ensemble with N_T trees, we take an average over all trees. This results in the following variable importance measure.

$$VI(X_j) = \frac{\sum_{i=1}^{N_T} VI(X_j, T)}{N_T} \tag{2.3}$$

2.2 Shrinkage Methods in Regression

Consider the usual regression problem where data is in the form of (x_i, y_i) for $i = 1, \cdots, N$, $x_i = (x_{i1}, x_{i2}, \cdots, x_{iM})'$. The linear regression model to predict target y_i has the form

$$f(x_i) = \beta_0 + \sum_{j=1}^{M} x_{ij}\beta_j \tag{2.4}$$

where β_j are the coefficients that form the elements of vector $\beta = (\beta_0, \beta_1, \cdots, \beta_M)'$. The ordinary least squares (OLS) estimates are obtained by minimizing the residual sum of squares for the target attribute. That is, minimizing

$$RSS(\beta) = \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2 \tag{2.5}$$

The OLS method provides unbiased coefficient estimates. For data with large number of attributes, predication accuracy can often be improved by sacrificing a little bias to reduce the variance of the estimates. This can be achieved through shrinking the coefficient estimates by imposing a penalty term. Such shrinkage methods are based on minimizing a penalized residual sum of squares (Friedman *et al.*, 2001). The least absolute shrinkage and selection operator (lasso) is a popular shrinkage method (Tibshirani, 1996). It is based on minimizing the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. This is equivalent to

$$\hat{\beta}^{lasso} = \underset{\beta}{argmin} \left\{ \frac{1}{2} \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{M} |\beta_j| \right\} \tag{2.6}$$

The λ parameter controls the penalization. This form of penalization results in some coefficients to be estimated as exactly zero, leading to interpretable models.

There is, however, no closed-form solution but efficient algorithms for computing the entire path of solutions as λ varies are available (Tibshirani, 1996).

2.3 Logistic Regression

The generalized linear model (GLM) (Myers *et al.*, 2012) describes the relationship between the mean of a target and input attributes where the target distribution is a member of the exponential family. A special case is logistic regression where the target attribute has only two possible values and is modeled as a Bernoulli random variable.

As before the training data is in the form of (x_i, y_i) for $i = 1, \cdots, N$, $x_i = (x_{i1}, x_{i2}, \cdots, x_{iM})'$ and target y_i has two possible values. Logistic regression constructs a classification model that assumes

$$E(y_i) = \frac{\exp(x_i'\beta)}{1 + \exp(x_i'\beta)} \tag{2.7}$$

where $\beta = (\beta_0, \beta_1 \cdots, \beta_M)'$. Maximum likelihood estimation is commonly used for parameter estimation in this method. References Hosmer Jr and Lemeshow (2004) and Myers *et al.* (2012) provide further details.

2.4 Likelihood Ratio Test

The likelihood ratio is a method for finding hypothesis test procedures (Casella and Berger, 1990). Assuming a random independent sample of size N, $y_1, y_2, \cdots, y_N$ with a pdf or pmf of $f(y \mid \theta)$, the likelihood function is

$$L(\theta \mid y_1, y_2, \cdots, y_N) = \prod_{i=1}^{N} f(y_i \mid \theta) \tag{2.8}$$

The likelihood ratio test (LRT) statistic for testing

$$H_0 : \theta \in \Theta_0 \tag{2.9}$$

$$H_1 : \theta \in \Theta_0^c \tag{2.10}$$

is the following

$$\gamma = \frac{\sup_{\Theta_0} L(\theta \mid X)}{\sup_{\Theta} L(\theta \mid X)} \tag{2.11}$$

where Θ denotes the entire parameter space. A LRT uses γ as the test statistic and rejects H_0 when $\gamma \leq k$, where k is determined by fixing type I error.

2.5 Control Charts

A control chart is a primary tool used for monitoring in statistical process control (SPC) (Montgomery, 1991). Figure 2.1 shows a typical control chart that plots a summary statistic of samples taken from a process versus time. In a simple case, the summary statistic could be the mean of the quality characteristics in samples taken from the process. This chart has three lines the center line (CL), lower control limit (LCL) and upper control limit (UCL) that convey where the summary statistic should fall in the absence of unusual variability. The idea is to use the control chart to monitor the process such that points outside the control limit convey unusual variability.

2.6 The Expectation-Maximization Algorithm

The expectation maximization (EM) algorithm is an iterative method for finding maximum likelihood estimates for statistical models that depend on unobserved latent variables (Dempster *et al.*, 1977). The idea is to alternate between an expectation step that estimates the latent variables and a maximization step that maximizes the

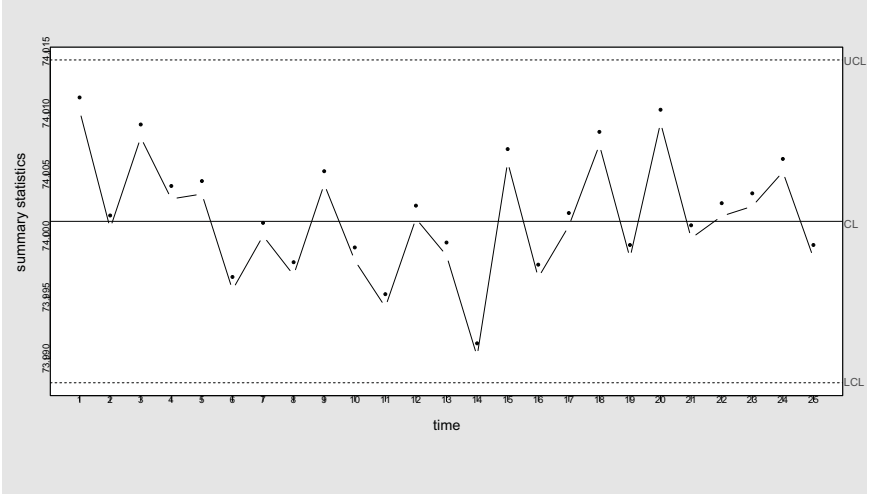


Figure 2.1: An Example Control Chart.

likelihood function based on the current estimates of the latent variables. The algorithm has various applications such as clustering, discriminate analysis and density estimation. We next describe it in the context of density estimation (Friedman *et al.*, 2001).

Consider a random variable Y whose distribution is a mixture of two Normal distributions such that $Y = (1 - Z)Y_1 + ZY_2$, where $Y_1 \sim N(\mu_1, \sigma_1^2)$ and $Y_2 \sim N(\mu_2, \sigma_2^2)$. The goal is density estimation for the two Normal distributions. Denoting the normal density with parameters μ_j, σ_j as $\phi_j(y, \theta)$, and using $Pr(Z = 1) = \pi$, the log-likelihood of N independent training instances can be written as

$$\sum_{i=1}^{N} \log[(1 - \pi)\phi_1(y_i, \theta) + \pi\phi_2(y_i, \theta)] \tag{2.12}$$

Direct maximization of Equation 2.12 is difficult due to the presence of the summation of the two Normal densities inside the logarithm. To overcome this, the EM algorithm considers unobserved latent variables z_i taking values 0 or 1 according to

$$z_i = \begin{cases} 0 & \text{if } y_i \sim N(\mu_1, \sigma_1^2) \\ 1 & \text{if } y_i \sim N(\mu_2, \sigma_2^2) \end{cases} \tag{2.13}$$

The log-likelihood is then written as

$$\sum_{i=1}^{N}[(1 - z_i)\log\phi_1(y_i) + z_i \log\phi_2(y_i)] + \sum_{i=1}^{N}[(1 - z_i)\log(1 - \pi) + z_i \log\pi] \tag{2.14}$$

Now, since the values of z_i are actually unknown an iterative method is used that substitutes the z_i's with their expected values

$$\zeta_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1 - \hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}(y_i)} \tag{2.15}$$

where

$$\hat{\pi} = \frac{\sum_{i=1}^{N}\zeta_i}{N} \tag{2.16}$$

from the previous iteration.

Starting with initial values for the parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2, \pi)$, an expectation step replaces the z_i values by their expected values in Equation 2.15. This is followed by maximizing the weighted log-likelihood function for obtaining updated parameter estimates. The iterations are continued until convergence.

2.7 Network Measures

A network is composed of a set vertices and edges. The topological structure of an example network, induced by its vertices and connecting edges, is depicted in Figure 2.2. Upon observing the topology, one might be interested to quantify its characteristics in order to answer question like what is the average number of edges that originate from the vertices? Such questions may be important for different tasks

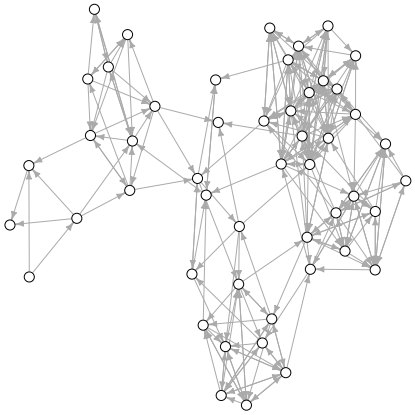


Figure 2.2: An Example Network.

such as comparing two networks. In this direction, many different network measures have been developed through the years (Freeman, 1979, 1977; Wasserman, 1994).

Network measures are generally extracted at both vertex and network level. The relative importance of a vertex within a network is captured through vertex level measures such as degree, closeness and betweenness. The degree of a vertex is simply the number of its adjacent edges, closeness is the number of edges needed to access every other vertex and betweenness is the number of geodesics (shortest paths) going through the vertex. It should be noted that such vertex level measures are sometimes averaged across the network to provide an overall measure for the whole network. Other network measures are captured at the network level and reflect the overall network topology. As an example, network density refers to the ratio of the number of edges and the number of possible edges.

Additional network measures may be captured through scan statics (Marchette, 2012; Neil *et al.*, 2014; Park *et al.*, 2013; Priebe *et al.*, 2005). The construction of the scan statistic involves enumerating fixed, defined, windows over the entire network. For example, Priebe *et al.* (2005) considered the window as kth-order neighborhoods

around each vertex. This neighborhood is defined as the subnetwork composed of the vertices that have a geodesics of length k or smaller to the vertex. A locality statistic, such as the number of edges in this neighborhood, is then calculated and a function, such as the maximum, of the locality statistic over all vertices is taken to be the scan statistic. Similarly, Neil *et al.* (2014) enumerated star and paths over the network for the construction of the scan statistic.

2.8 Statistical Network Models

There exists a large literature on the statistical modeling of networks. Early work focused on modeling the observed set of edges in a single snapshot of the network (static network). The simplest of these is the Erdos-Renyi random graph model that describes networks where edges are formed independently between each pair of vertices with a common probability (Erdos and Renyi, 1959). This is an overly simple model and various attempts have been made to model systematic deviations from pure randomness (Frank and Strauss, 1986; Hoff *et al.*, 2002; Wang and Wong, 1987). As an example, the stochastic blockmodels (SBM) (Wang and Wong, 1987) is a multi-class extension of the Erdos-Renyi model. This model assigns a class to each vertex and uses a different edge probability for each pair of classes.

Most real networks have dynamic components. For example, in a social network, edges may be added or deleted at any time. The static models fail to model the underlying temporal dynamics. Perhaps, the simplest model for network dynamics is to view the Erdos-Renyi as a dynamic network that starts with the unconnected set of vertices and adds a different edge to the network with fixed probability at each subsequent time. Also in this direction, Barabási and Albert (1999) presented a preferential attachment model. This model starts with a set of unconnected vertices, adding a vertex at each time stamp that forms edges with the existing vertices. The

probability that the new vertex forms an edge with an existing vertex is modeled as a function of the existing vertex's degree. Other work like Leskovec *et al.* (2007), Chakrabarti *et al.* (2004), Pennock *et al.* (2002) also present graph generation models that result in networks with known network properties.

A more recent area of interest is change detection in network streams. As an example, McCulloh and Carley (2011) constructed control charts over different network statistics over time. The work by Priebe *et al.* (2005) and Marchette (2012) monitored scan statistics for this purpose. Similarly, the work by Park *et al.* (2013) used a fusion of network statistics (including the scan static) to detect changes in a stream of networks.

Chapter 3

MULTI-TARGET ENSEMBLE

3.1 Introduction

Many machine learning algorithms predict a single target attribute using a set of input attributes. Many real world problems, however, involve the prediction of several target attributes using a common set of input attributes. Problems of this type arise naturally in many fields such as manufacturing: to predict various quality aspects of a product using the manufacturing settings (MacGregor *et al.*, 1994), marketing: to predict different aspects of consumer behavior based on consumer characteristics (Zhang *et al.*, 2005), environmental sciences: to predict the distribution of several species using environmental conditions (De'Ath, 2002; Demšar *et al.*, 2006) and education: to predict different learning outcomes based on educational and demographic attributes (Azarnoush *et al.*, 2013; Tatsuoka and Lohnes, 1988). A typical solution to such problems is the independent construction of models for the prediction of each target attribute. However, alternative approaches that leverage multiple target attributes may be pursued (Blockeel *et al.*, 1998; Caruana, 1998).

Tree models are widely used learning algorithms that recursively partition the instances at the nodes into homogeneous child nodes. As an example, classification and regression trees (CART) (Breiman *et al.*, 1984) select partitions to minimize the Gini index for classification problems and to minimize the sum of squares for regression problems. The homogeneity is evaluated with respect to a single target attribute. Furthermore, collections of trees have been used towards constructing ensembles. The construction of such ensembles involves injecting some form of perturbation in train-

ing. The goal is to construct a collection of diverse, yet strong base learners and the final prediction is formed from a summary over them. Bagging (Breiman, 1996) and random forest (RF) (Breiman, 2001) provided effective examples of such methods.

Tree models have been extended for problems with multiple target attributes. As with single target attribute problems, the construction of these models involves the partitioning of the instances at the nodes to the most homogeneous child nodes. The homogeneity is, however, measured with respect to all of the target attributes. For example, Caruana (1993) selected partitions to minimize the average entropy over the target attributes. Later, Caruana (1997) proposed partitions to minimize a weighted average of entropies across the target attributes. Similarly, Blockeel *et al.* (1998) selected partitions to minimize the sum of entropies (classification) or the sum of variances (regression) of the individual target attributes. Additionally, De'Ath (2002) selected partitions to minimize the total sum of squares of the target attributes. Ensemble methods have also been extended for problems with multiple target attributes. For example, Kocev *et al.* (2007), Kocev *et al.* (2013) and Aho *et al.* (2012) constructed ensembles where the learners are the tree models for multiple target attributes proposed by Blockeel *et al.* (1998). Similarly, Segal and Xiao (2011) constructed ensembles of the trees proposed by De'Ath (2002).

The cited literature on tree models for the multi-target problem select partitions to minimize the impurity of the child nodes, where the impurity is measured using the multiple target attributes. A shortcoming of such an approach is the possible disagreement across the target attributes in selecting the optimal split. As the number of target attributes increases, fewer splits in a tree will be optimal for any one target attribute. A modest exception was to select partitions to minimize a weighted average of entropies across the target attributes (Caruana, 1997). In theory, this can allow for splits to favor a specific target attribute and, thus, overcome the mentioned

shortcoming. However, an appropriate weight function is needed for each split. A method, based on steepest descent hill climbing, was outlined to learn the weights for each target attribute without providing further results.

Here, we present a new method for a tree-based ensemble that leverages from multiple target attributes called the compound forest (CF). The basic idea is to construct a tree with splits based on one target and use the constructed regions to obtain predictions for another target attribute. Specifically, given a numerical target attribute τ (regression problem), a tree can be considered as a partition of the feature space into rectangular regions (for numerical predictors) with the prediction equal to the average of τ values of the instances in a region. With multiple target attributes, this process can be separated. That is, one can construct a tree from splits based on a target attribute y, and, thereby, obtain a partition. A prediction can be generated for target attribute τ from the average of τ values of the instances in each region of the partition.

As with most ensemble methods, the CF method involves injecting perturbations in training. By training the base learners using different target attributes, the approach exploits the multiple target attributes for the perturbations. The boundaries between homogeneous regions for different, yet related target attributes, are expected to be similar. Further perturbations include different data samples and splits from a randomly selected subsets of input attributes. By constructing each tree in a randomly selected subspace of the feature space, and selecting the useful trees for the final prediction (through solving a regularized regression problem, as explained later), the method is essentially performing a random subspace search (Ho, 1998) for regions that are homogeneous with respect to different target attributes.

Due to the high diversity of base learners, the ensemble likely consists of relevant as well as non-relevant members for the prediction of a specific target attribute τ. A

common approach to aggregate predictions across ensembles is averaging with equal weights to each base learner. We note, however, that in the case of highly diverse base learners with different relevance, simple averaging of the learners can degrade accuracy as the effect of relevant base learners may be diminished by the presence of highly non-relevant ones (Friedman and Popescu, 2003). To make the ensemble robust, the CF assigns weights through solving a regularized regression problem that takes the relevance of each base learner with respect to τ into account. The weight assignment introduces sparsity among the base learners by shrinking the weights of the non-relevant base learners to zero.

The rest of this chapter is arranged as follows. Section 3.2 discusses the connection of CF to the existing literature. Section 3.3 presents a detailed explanation of the method. CF is explained in the context of predicting a single target attribute τ in the presence of $y_t, t = 1, \ldots, T$. Section 3.3.3 considers the predictions of all target attributes. Sections 3.4 summarizes the results of experiments with synthetic and real data and includes comparisons to other related models. Finally, Section 3.5 concludes the chapter and provides directions for future research.

3.2 Related Work

A multi-label problem with a number of binary target attributes was considered by Zhang *et al.* (2005). Ensembles of classification trees, trained in the traditional manner, were formed for each target attribute and a subset of these trees was selected to predict a target attribute. Also, Breiman and Friedman (2002) separately trained models towards the prediction of multiple target attributes. Here, linear, ordinary least squares models for different target attributes were obtained, and a linear combination of these models was used to predict each target attribute. A similarity of these references to the CF is the sharing of models across multiple target attributes.

A key difference, however, lies in the form of knowledge transfer. In the works by Zhang *et al.* (2005) and Breiman and Friedman (2002), each base learner provides a prediction only for the target attribute on which it is trained. In CF, each base learner provides a different prediction based on the target attribute of interest and a target-specific linear combination is used as the final prediction.

In another direction, Breiman (2000) introduced *artificial* variability to the target attribute, referred to as *output smearing*, to improve the generalization performance of an ensemble. However, only a single target problem was considered. Perturbed training sets are produced by adding random variation to the target attribute, and an ensemble of base learners are constructed. That is, although the final goal is the prediction of τ, a tree in the ensemble is trained on (and predicts) y where $y = \tau + \epsilon$ and ϵ denotes a random Gaussian noise term. Similarly, CF exploits the availability of multiple target attributes to introduce variability in the ensemble construction by varying the target attributes used to train the base learners (training on y to predict τ).

The CF also has connections to adaptive nearest neighbor (ANN) methods (Hastie and Tibshirani, 1996). In general, nearest neighbor methods assume target attribute values are roughly constant within neighborhoods. Given a test instance, the prediction is obtained from the target attribute values of training instances within its neighborhood. The neighborhood is determined through a distance measure that quantifies the *closeness* of the test instance to the training instances. The ANN techniques adjust the distance measure so that the resulting neighborhoods are extended in directions with small variation in the target attribute values. The work by Thrun and O'Sullivan (1996) uses the ANN technique for the multi-task problem. The CF method shares some commonalities to this approach, explained by the connection of RF to the ANN methods (Lin and Jeon, 2006). In this view, the forest creates a

unique distance measure for each instance and then fits a weighted nearest neighbor model. That is, the predictions from the RF are weighted averages of the training instances, where the assigned weights are based upon a distance measure created by the ensemble of the trees that captures the closeness of the test instance to the training instances. To draw the connection, we note that the CF method trains the trees of an ensemble on different target attributes, which is similar to adjusting the distance measure based on different target attributes in Thrun and O'Sullivan (1996), and then reuses it for the prediction of each target attribute. The adjustment in Thrun and O'Sullivan (1996) is done through a ANN method that adjusts a weighted Euclidean distance measure so that the resulting neighborhoods are extended in directions for which the y is roughly constant. This adjusted measure is then used for the classification of target attribute τ. That is, for each test instance, its closest neighbors are determined using y and the τ values of the neighbors are used to predict τ for the test instance. In the CF method, the distance measure is adjusted locally using trees. Each tree finds the closest neighbors of a test instance using y (the training instances that fall in the same terminal as the test instance in a tree trained using y), and the τ values of these neighbors are used for the prediction of τ for the test instance.

Furthermore, CF can be regarded as an extension of the importance sampled learning ensemble (ISLE) framework (Friedman and Popescu, 2003), which considers only single target attribute problems, to multi-target problems. This framework describes many well known ensemble methods in the context of random Monte Carlo integration methods based on different importance sampling strategies. As with all supervised learning problems, the goal is to predict the target attribute y given the vector of input attributes x with a joint probability distribution $z(x, y)$. Each base learner, $f(x, \theta)$, is a function of the input attributes and a set of parameters $\theta \in \Theta$. These parameters define the prediction model for the target attribute. These are

combined through a linear model of the form

$$F(x) = a_0 + \int_\theta a(\theta) f(x, \theta) d\theta \tag{3.1}$$

where $a(\theta)$ is the corresponding coefficient in the linear model.

Towards forming the base learners, numerical quadrature rules are employed to find a collection of M evaluation points, $\{\theta^m\}_1^M$, and

$$F(x) \approx c_0 + \sum_{m=1}^{M} c_m f(x, \theta^m) \tag{3.2}$$

is used to approximate $F(x)$ where c_m is the corresponding weight applied to the mth base learner. Importance sampling that randomly draws a collection of evaluation points from a probability distribution $r(\theta)$ is employed for this. This distribution should assign higher probability to evaluation points that are more relevant for approximating $F(x)$. A possible measure of the (lack of) relevance of an evaluation point θ is the prediction risk of using θ alone in a single point rule ($M = 1$). This measure is

$$W(\theta) = \min_{\alpha_0, \alpha} E_{z(x,y)} L(y, \alpha_0 + \alpha f(x, \theta)) \tag{3.3}$$

where $L(\cdot)$ is the loss function.

Finding and sampling from an appropriate probability distribution $r(\theta)$ that assigns higher probability to points that are more relevant for approximating $F(x)$ (θ's with smaller $W(\theta)$) is problem specific. However, the process can be approximated by repeated perturbation of some aspect of the problem and finding the θ of the optimal single point rule that minimizes Equation 3.3.

We note that the use of a single point rule is the same as using a single model (for example, a single tree) for the prediction of y. The θ of the optimal single point rule is

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} W(\theta). \tag{3.4}$$

In this process, the mth base learner, $f(x, \theta^m)$, is formed by perturbing the problem and finding θ^m (θ^* for the mth perturbed problem). This is repeated M times and the collection of base learners, $\{f(x, \theta^m)\}_1^M$, is used to form the quadrature rule in Equation 3.2. The second problem in ISLE is determining the quadrature coefficients $\{c^m\}_0^M$. This can be considered as a regression problem where y is the target attribute and the predictors are the base learners $\{f(x, \theta^m)\}_1^M$.

An example of an ensemble method that fits in this framework is RF. Each base learner, $f(x, \theta^m)$, is a decision tree with parameter θ which is the partition of the feature space imposed by a tree (defined by the split attributes and split values) and the assigned values in the terminal nodes. Therefore, $r(\theta)$ should assign higher probability to trees with partitions of the feature space (θ) that are more relevant for predicting y (more homogeneous with respect to y). The perturbation in RF involves the modification of the joint distribution $z(x, y)$ to $z^m(x, y)$ by constructing each tree on a different bootstrap sample drawn from the data. Another aspect of the perturbation is the modification of the algorithm by selecting the optimal split among a randomly chosen subset of the input attributes at each node. This hybrid perturbation allows for the construction of different trees that all address the same problem of predicting y. The final prediction is a linear combination with equal weights $\{c^m\}_0^M$ for each tree.

The sampling probability distribution $r(\theta)$ is characterized by its location and dispersion. These should be chosen appropriately so that most of its mass is placed in regions where the integrand (Equation 3.1) realizes important values and relatively

small mass elsewhere. The location of $r(\theta)$ should be near θ^* of the unperturbed problem, and the dispersion is determined by the injected perturbation that controls how differently the prediction problem is tackled by each base learner. The dispersion of $r(\theta)$ is related to the trade-off between the strength and the correlation of individual base learners. It is shown that good results are achieved with an ensemble of moderately strong and low-correlated base learners (Breiman, 2001). The CF method introduces a novel approach towards forming a sampling probability distribution by taking advantage of the availability of multiple target attributes to perturb the problem. Specifically, one aspect of the perturbation is altering the target attribute y in Equation 3.3 for constructing the base learners. Then, as with the ISLE, the optimal single point rule (θ^*) is found for each perturbed problem which is used in forming the base learners. The base learners differ with respect to the target attribute that they were trained on which imposes a *grouping* among them. Namely, base learners trained using one target attribute will be in the same group. This grouping is incorporated for learning the quadrature coefficients $\{c^m\}_0^M$ by employing a sparse group regression model.

3.3 Compound Forest

In the problem under study, each instance is in the form of (x_i, y_i) with J input attributes $x_i = (x_{i1}, \ldots, x_{iJ})$ and T numerical target attributes $y_i = (y_{i1}, \ldots, y_{iT})$ with joint distribution $z(x, y)$. We have access to a random sample of size N from $z(x, y)$, which represents an empirical point mass approximating the joint distribution. Without loss of generality, the goal here is to predict any target attribute from the T target attributes y_i, which we denote by τ, using the input attributes and by leveraging all the target attributes. Section 3.3.3 extends this by discussing the prediction of all target attributes.

The CF method has two main steps of forming a collection of base learners and then obtaining a final prediction using a linear combination of these base learners. To give a high level description of the method, these two steps are briefly discussed below. This is followed by a detailed description. The last subsections discuss the prediction of multiple target attributes and the computational complexity of the method.

The base learners in CF are derived from decision trees. For each tree, the problem is perturbed by modifying the algorithm and the joint distribution. The availability of the multiple target attributes allows for a new modification of the algorithm by selecting different target attributes to train the trees on. Additionally, the split is selected from a randomly selected subset of input attributes and each tree is constructed using a different data sample. The resulting trees correspond to partitions of the feature space that generate predictions for τ using the average of τ values of the instances in each region of the partition. These predictions form the base learners.

The perturbation determines the diversity of the base learners, with high diversity leading to the existence of relevant as well as non-relevant base learners in the ensemble. Simple averaging will likely degrade the prediction as both relevant and non-relevant base learners are assigned equal weight in the final prediction. It is, therefore, desirable to introduce sparsity in the base learners by shrinking the weights of the non-relevant base leaner to zero. In this direction, the work on single target attribute prediction by Friedman and Popescu (2003) adopts the l_1 penalty to introduce sparsity and determines weights accordingly. For the CF method, we introduce sparsity between and within groups of base learners trained on each target attribute and assign weights accordingly. The relevancy of the base learner in predicting τ is determined by the target attribute, the selected input attributes at each split and the instances that are used in its training. That is, the set of base learners trained on a target attribute highly related to τ will likely be moderately good in predicting

τ. Whereas, those base learners trained on an unrelated target attribute, will likely be poor. Still, it is not expected that every base learner trained on a related target attribute is useful in the final prediction. Therefore, sparsity between and within groups of base learners trained on each target attribute is desirable. Towards this end, a sparse group regression model that integrates the l_1 and l_2 penalties to have the desired between and within group sparsity effect has been adopted in the weight assignment of the base learners.

3.3.1 Base Learner Formation

Each base learner in the compound forest is a tree that partitions the feature space into regions to predict target attribute τ. These regions correspond to the terminal nodes of the tree, and a different prediction is given for each terminal node. A splitting criterion on input attribute x_j is denoted by $\delta(x_j)$ which results in the partition of the feature space into two regions of $R_1(x_j)$ and $R_2(x_j)$ and constants $\kappa_1(y)$ and $\kappa_2(y)$ are assigned to each region.

Towards the construction of the mth tree, the joint distribution $z(x, y)$ is modified to $z^m(x, y)$ by drawing a different sample. A target attribute is then selected, denoted by y_{t_m}, and the tree is constructed from splits based on this target attribute. At each node of the tree, a subset of the attributes is randomly selected, and the attribute and the splitting criterion that minimize

$$\min_{\delta(x_j)} \left[\min_{\kappa_1(y_{t_m})} \sum_{x_i \in R_1(x_j)}^{\cdot} (y_{it_m} - \kappa_1(y_{t_m}))^2 + \min_{\kappa_2(y_{t_m})} \sum_{x_i \in R_2(x_j)}^{\cdot} (y_{it_m} - \kappa_2(y_{t_m}))^2 \right] \tag{3.5}$$

are chosen. Here, $\kappa_1(y_{t_m})$ and $\kappa_2(y_{t_m})$ are taken to be the average of the y_{t_m} values of the instances that occupy $R_1(x_j)$ and $R_2(x_j)$. This process is continued until some stopping rule is met.

The mth constructed tree has V^m terminal nodes, each corresponding to a region

in the feature space denoted by $R_v^m, v = 1, \ldots, V^m$. We further denote the average of τ values of the instances that occupy R_v^m by $\kappa_v^m(\tau)$. Note that the mth tree is trained on target attribute y_{t_m}. However, the tree is used to provide a prediction for target attribute τ. Region R_v^m generates a prediction for τ which is taken to be $\kappa_v^m(\tau)$. The tree's prediction is

$$f(x, \theta^m, y_{t_m}, \tau) = \sum_{v=1}^{V^m} \kappa_v^m(\tau) I\left(x \in R_v^m\right) \tag{3.6}$$

where $I\left(x \in R_v^m\right)$ is an indicator function denoting the presence of instance x in R_v^m. We note that $\theta^m = \{R_v^m, v = 1, \ldots, V^m\}$ for the trees in CF and the terminal node predictions are obtained using target attribute τ.

Different approaches may be pursued for setting the number of trees per target attribute. This may include approaches that select an optimal number of trees for each target attribute. In our implementation, however, a simple approach of using equal number of trees per target attribute is used. That is, denoting the number of trees per target attribute by M_t, we set M_t to equal a constant q. This results in a ensemble with a total of $T \times q$ trees. For the construction of the mth tree, y_{t_m} is selected according to

$$y_{t_m} = y_1 I(m \in [1, q]) + y_2 I(m \in [q+1, 2q]) + \cdots + y_T I(m \in [(T-1)q+1, Tq]) \tag{3.7}$$

where $I(m \in [a, b])$ is an indicator function that m is the $[a, b]$ interval.

3.3.2 Linear Combination Formation

Note that the formation of $\{f(x, \theta^m, y_{t_m}, \tau)\}_1^M$ can be regarded as a transformation of the J dimensional feature space $x = (x_1, \ldots, x_J)$ to a new M dimensional feature space $\phi(\tau) = (\phi^1(\tau), \ldots, \phi^M(\tau)) = (f(x, \theta^1, y_{t_1}, \tau), \ldots, f(x, \theta^M, y_{t_M}, \tau))$. The new feature space likely consists of relevant as well as non-relevant features with regard to predicting τ. For the final prediction, a summary of these features is re-

quired. Toward this end, rather than assigning equal weights to each feature (as done in ensemble learning methods such as bagging and RF), it is more reasonable to perform a supervised post-processing that takes into account each feature's relevance for predicting τ in the weight assignment (Equation 5.7).

The weight assignment involves solving the regression problem where τ is the target attribute, and the predictors are the base learners $\{f(x, \theta^m, y_{t_m}, \tau)\}_1^M$. Because a subset of the base learners are trained on target attribute $y_t, t = 1, 2, \ldots, T$, this may be regarded as a form of grouping over $\phi(\tau)$. There are, therefore, a total of T groups, each of length M_t, where $M = \sum_{t=1}^{T} M_t$. As discussed in Section 3.3.1, our implementation uses $M_t = q$, where q is constant across all target attributes. If two base learners, $f(x, \theta^m, y_{t_m}, \tau)$ and $f(x, \theta^{m'}, y_{t_{m'}}, \tau)$, are trained on the same target attribute y_t (i.e. $y_{t_m} = y_{t_{m'}} = y_t$), then they are in group $\phi^{(t)}(\tau)$.

Incorporating this natural grouping in the assignment of weights for the final prediction may lead to higher accuracy. Due to the nature of the generation of these features, it is reasonable to assume sparse effects both on a group and within group level. The sparsity on the group level can be explained through the fact that not all target attributes $y_t, t = 1, ..., T$, used in training the base learners, are expected to be relevant in predicting the τ. Hence, the coefficients placed on the group of base learners extracted from trees trained on the unrelated target attributes should be shrunk toward zero. This increases the robustness of the algorithm in the of presence of unrelated target attributes as it insures selective transfer of knowledge (Thrun and O'Sullivan, 1996). On the other hand, the within group sparsity is expected due to the diversity of the base learners trained on a single target attribute. That is, even within a single group of base learners trained on one target attribute, diversity is likely to be incurred due to the sampling of the training instances and features. As a result, the committee of base learners trained on a single target attribute will also

likely consists of both relevant as well as non-relevant base learners.

A method is used that introduces sparsity both in the group and within the group. The sparse group lasso (SGL) (Friedman *et al.*, 2010; Simon *et al.*, 2013) integrates the l_1 and l_2 penalties to have the desired group and within group sparsity effect. Denoting the $M_t \times 1$ weight vector of group t by $c^{(t)}(\tau) = \{c^m(\tau) \mid y_{t_m} = y_t\}$ and the entire weight vector $c(\tau) = \left(c^1(\tau), ..., c^M(\tau)\right) = \left(c^{(1)}(\tau), ..., c^{(T)}(\tau)\right)$, the regularized regression problem is

$$\min_{c(\tau)} \frac{1}{2N} \left\| \tau - \left\langle \phi^{(t)}(\tau), c^{(t)}(\tau) \right\rangle \right\|_2^2 + (1-\gamma)\lambda \sum_{t=1}^{T} \sqrt{M_t} \left\| c^{(t)}(\tau) \right\|_2 + \gamma\lambda \left\| c(\tau) \right\|_1 \quad (3.8)$$

The two meta parameters $\gamma \in [0, 1]$ and $\lambda \geq 0$ control the sparsity of the solution. In the two extremes, $\gamma = 0$ provides the group lasso fit (Yuan and Lin, 2005) and $\gamma = 1$ provides the lasso fit (Tibshirani, 1996). In order to consider different amounts of regularization, a similar approach to Simon *et al.* (2013) is used in which values for γ are fixed and solutions for a path of values for λ is computed. The path starts from a λ value that is the smallest value such that all coefficients are shrunk to zero and is continued by decreasing λ until near an un-regularized solution. After the assignment of weights through solving Equation 3.8, the compound forest prediction for instance x on target attribute τ is

$$\sum_{m=1}^{M} c^m(\tau)\phi^m(\tau). \quad (3.9)$$

This is a linear combination of predictions for target attribute τ generated from partitions obtained using target attributes $y_t, t = 1, ..., T$. The weight assignment takes the relevance of these predictions into account to guard against highly non-relevant predictions. We note that our implementation restricts the maximum number of terminal nodes to six. This is due to the findings in Friedman and Popescu (2003) that report the benefit of shallow trees when regularization is used in weight assignment. Preliminary experiments with other values (e.g., 10) indicate little change in

the performance of CF. The complete algorithm for CF is summarized in Algorithm 1.

Algorithm 1: Compound Forest

for m=1:M

1. Modify the joint distribution to $z^m(x, y)$.

2. Select a target attribute y_{t_m} according to $y_{t_m} = y_1 I(m \in [1, q]) + y_2 I(m \in [q+1, 2q]) + \cdots + y_T I(m \in [(T-1)q + 1, Tq])$.

3. Construct a regression tree whose splits are based on y_{t_m}. At each node of the tree, the split is chosen from a randomly selected subset of input attributes.

4. Use the tree to predict τ by

$$\phi^m(\tau) = \sum_{v=1}^{V^m} \kappa_v^m(\tau) I\left(x \in R_v^m\right)$$

end

5. Assign weights to each tree by solving the regularized regression problem

$$\min_{c(\tau)} \frac{1}{2N} \left\| \tau - \left\langle \phi^{(t)}(\tau), c^{(t)}(\tau) \right\rangle \right\|_2^2 + (1-\gamma)\lambda \sum_{t=1}^{T} \sqrt{M_t} \left\| c^{(t)}(\tau) \right\|_2 + \gamma\lambda \left\| c(\tau) \right\|_1$$

6 Form the final prediction by

$$\sum_{m=1}^{M} c^m(\tau) \phi^m(\tau).$$

3.3.3 Predicting Multiple Target Attributes

It should be noted that although CF is described in the context of predicting a single target attribute τ, the partitions of the feature space obtained from the ensemble may be used for the prediction of all target attributes. That is, an ensemble

of trees is constructed through steps 1-3 in Algorithm 1, and regardless of the target attribute used in training, each tree corresponds to a different partition of the feature space. The resulting partitions, $\{R_v^m, v = 1, \ldots, V^m\}_1^M$, can be used to generate a prediction for each target attribute, $y_t, t = 1, \ldots, T$. This is taken to be the average of y_t values of the instances that occupy R_v^m, denoted by $\kappa_v^m(y_t)$. As in Equation 3.6, the mth tree's prediction for target attribute y_t is

$$f(x, \theta^m, y_{t_m}, y_t) = \sum_{v=1}^{V^m} \kappa_v^m(y_t) I\left(x \in R_v^m\right). \tag{3.10}$$

The ensembles's prediction for y_t is then formed from a linear combination of the trees' predictions obtained by solving Equation 3.8 for y_t.

This sharing of the ensemble reduces the computations when the final goal is the prediction of more than a single target attribute. Furthermore, in the case of distributed data with target attribute y_t in location t, the trees trained on each target attribute may be trained locally so that $\left\{\{R_v^m, v = 1, \ldots, V^m\}_1^M \mid y_{t_m} = y_t\right\}$ is obtained from location t. Then $\left\{\{R_v^m, v = 1, \ldots, V^m\}_1^M \mid y_{t_m} = y_t\right\}_1^T$ is shared centrally to be used in predicting each target attribute.

3.3.4 Computational Complexity

The computational complexity of CF is evaluated in terms of its two steps. The formation of base learners can be broken down to the construction of M trees which is $O\left(MJ'\nu \log\left(\nu\right)\right)$, where ν is the number of instances used to train each tree of depth $\log(\nu)$ and J' is the number of attributes used at each node (Witten and Frank, 2005). We note that the construction of trees may be parallelized. Then the ensemble generates predictions for τ for each instance which is $O\left(M \log\left(\nu\right)\right)$.

The linear combination of base learners is formed using the SGL (Simon *et al.*, 2013). For each group with M_t base learners, an accelerated gradient is performed

which is $O(NM_t)$ per iteration. For a convergence threshold of ω, in the worst case scenario, an accelerated gradient descent with restarts takes $O(1/\sqrt{\omega})$ iterations (Nesterov, 2007). This process is then cyclically repeated through the groups.

As shown in the experimental evaluation, the trees in CF may be shallow with fast look-up time for prediction. Furthermore, the use of SGL introduces sparsity, assigning zero weight to some base learners. The corresponding base learners need not be evaluated for prediction. These elements promote fast predictions.

3.4 Experimental Evaluation

CF is compared to three other related methods. This comparison includes the RF (Breiman, 2001) and a modified version of RF that forms the final prediction of the ensemble through the lasso method (Friedman and Popescu, 2003)(referred to as ISRF). These two methods do not use the multiple target attributes in forming the ensemble and so a separate model is constructed for each target attribute. Our comparison further includes the multi-target random forest (Kocev *et al.*, 2013) (referred to as MTRF). Experiments are implemented in R 3.0.3 Software on a Windows 7 Enterprise Intel Core i7-3770 CPU (3.4 GHz) 64bit Operating System.

All the methods in the comparison are based on the RF methodology and use $\lfloor J/3 \rfloor$ input attributes at each node during training (Friedman *et al.*, 2001). The ensemble prediction for all methods are formed through a linear combination of base learner predictions. Equal weights are assigned to each base learner in RF and MTRF, whereas the weights in ISRF and CF are determined through a post-processing step that involves solving a regularized regression problem. For these two methods, the maximum number of terminal nodes is restricted to six constructed on 50% of the training data selected without replacement. The number of terminal nodes is restricted due to the findings in Friedman and Popescu (2003) that report the benefit

Data set Name	Size	Input Attributes	Target Attributes
Synth 1	1000	40	5
Synth 2	1000	40	20
Synth 3	1000	40	20
SARCOS	48933	21	7
CS	400	11	3
BG	136	7	4
HSB	600	9	5
LDP	50	14	5

Table 3.1: Data Set Description.

of post-processing in the case of shallow trees where six is used.

In order to consider different amounts of regularization in the weight assignment of CF, γ of 0, 0.05, 0.55, 0.7, 0.95, and 1 are considered. For each γ value, solutions for a path of λ values are computed. The path starts from a λ value that is the smallest value such that all coefficients are shrunk to zero, denoted as $\lambda_{\max}$, and is continued by decreasing λ until near an un-regularized solution of $0.01\lambda_{\max}$.

The methods are compared based on eight data sets. The first three data sets are simulated with known characteristics. The next five data sets are real data obtained from different domains. Table 3.1 provides a summary of the data sets. Five-fold cross validation is used on data sets with less than 1000 instances. For larger data sets, a sample of 350 is used for training and a sample of 500 for testing which is replicated five times. For CF and ISRF, that involve a post-processing step, one forth of the training fold is used for validation (for smaller data sets) and a sample of 100 instances from the training set is used as validation (for larger data sets).

For evaluation, the relative root mean squared error (RRMSE) on each target

attribute is considered. The RRMSE of method for target attribute y_t is defined as

$$RRMSE_t = \sqrt{\frac{\sum(\hat{y}_{it} - y_{it})^2}{\sum(\hat{y}_{it} - \bar{y})^2}} \tag{3.11}$$

For comparison, the comparative relative root mean squared error (CRRMSE) on each target attribute is considered (Friedman and Popescu, 2003). This is defined as

$$CRRMSE_t^d = \frac{RRMSE_t^d}{\underset{g}{min} RRMSE_t^g} \tag{3.12}$$

which is the ratio of the RRMSE for target attribute y_t of method d to the RRMSE of the best method being compared with on a particular data set. The best method, hence, receives a value of 1 and others have larger values. Results are presented for each data set individually, and then final significance tests are conducted to compare CF to competitors.

To set the number of trees in the ensembles for the comparison, the RRMSE of ensembles of different sizes for CF, ISRF and RF are considered. The same number of trees per target attribute is used for CF (i.e. M_t=q, $t = 1, \cdots, T$) which results in $T \times q$ trees in CF. The same number of trees are used for the other competitors. Figure 3.1 depicts the RRMSE for different values of q for, without loss of generality, the first target attributes in four selected data sets for CF, ISRF and RF. As can be observed, results are stable after $q = 100$ for the three methods. Furthermore, it is shown that the performance of MTRF is stable after 50 trees are added (Kocev *et al.*, 2013). Therefore, we set $q = 100$ so that the ensembles each consist of $100T$ trees. Note that this value will always be larger than 50 so that results for MTRF are stable. Under this set up, a data set with T target attributes requires $T \times q$ trees for CF and MTRF that are used across all target attributes, while the same data set requires $T \times q$ trees per target attribute for ISRF and RF resulting in a total of $T^2 \times q$ trees for all target attributes because separate ensembles are constructed for

each target attribute in these two methods.

3.4.1 Synthetic Data

To evaluate the CF method, it is desirable to control a number of properties of the data: the incorporation of nonlinearity, the control of the relevancy of the target attributes to each other, the number of target attributes, and the knowledge of the true function. The random function generator in Friedman (2001) has been modified to meet these criteria. This generator was also used in Friedman and Popescu (2003) for evaluating the univariate ISLE.

Each target function is in the form of

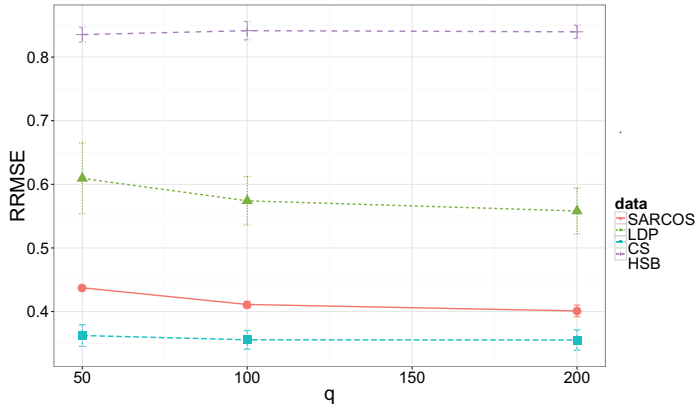
$$F_t^*(x) = \sum_{l=1}^{L} a_{lt} h_l(x), t = 1, ...T, \tag{3.13}$$

where the coefficients a_{lt} are randomly generated from a uniform distribution $U[0, 1]$ and $L = 20$ here. Each $h_l(x_l)$ is a function of a randomly selected subset of the attributes. The size of each subset, n_l, is randomly generated from $\lfloor 1.5 + e \rfloor$, where e is generated from an exponential distribution with mean 2. Each $h_l(x_l)$ is then taken to be an n_l-dimensional Gaussian function and target attribute y_t for instance i is taken to be

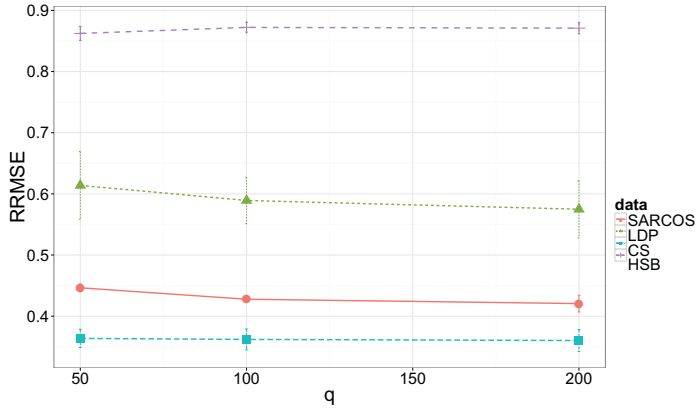
$$y_{it} = F_t^*(x_i) + \varepsilon_{it} \tag{3.14}$$

where ε_{it} is generated from a Gaussian distribution with standard deviation of $\eta\sigma$ where σ denotes the standard deviation of $F_t^*(x_i)$.

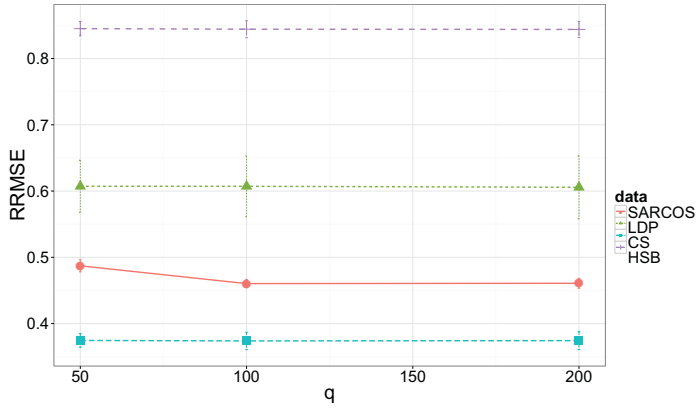
It should be noted that except for η, the parameters used in our experiment are those used in Friedman and Popescu (2003), where single target attribute problems were considered and $\eta = 1$ was used. This was modified in our experiment because η allows us to control the relationship between the target attributes with smaller values inducing higher relevancy to each other.



(a)



(b)



(c)

Figure 3.1: The RRMSE of Selected Target Attributes Versus q Over Five Replicates. Part (a), (b) and (c) Depict Results for CF, ISRF and RF, Respectively. Results Are Stable After $q = 100$.

An experiment with data with different number of target attributes with different relevance to each other is conducted. Three different cases are considered. Table 3.2 summarizes the considered data sets. Since all target attributes are generated under an identical distribution, we consider only one target attribute in our experiments with the synthetic data (the first target attribute is selected without loss of generality).

Data Set Name	Number of Input Attributes	Number of Target Attributes	η
Synth 1	40	5	0.1
Synth 2	40	20	0.1
Synth 3	40	20	0.9

Table 3.2: Synthetic Data Sets Descriptions.

Synth 1

The first synthetic data consists of 40 input attributes and five related target attributes ($\eta = 0.1$). This data set presents a case in which there is moderate amount of useful information to be shared across the target attributes since the five target attributes are related. Figure 3.2 depicts the CRRMSE of the different methods over five replicates. As can be observed, CF takes advantage of the related target attributes to improve performance.

Synth 2

The second synthetic data consist of 40 input attributes and 20 related target attributes ($\eta = 0.1$). This data set presents an example in which there is a large number of related target attributes and, hence, useful information is shared across the target attributes to improve prediction. Figure 3.3 depicts the CRRMSE of the different methods over five replicates. As can be observed, CF takes advantage of the large number of related target attributes for performance improvement.

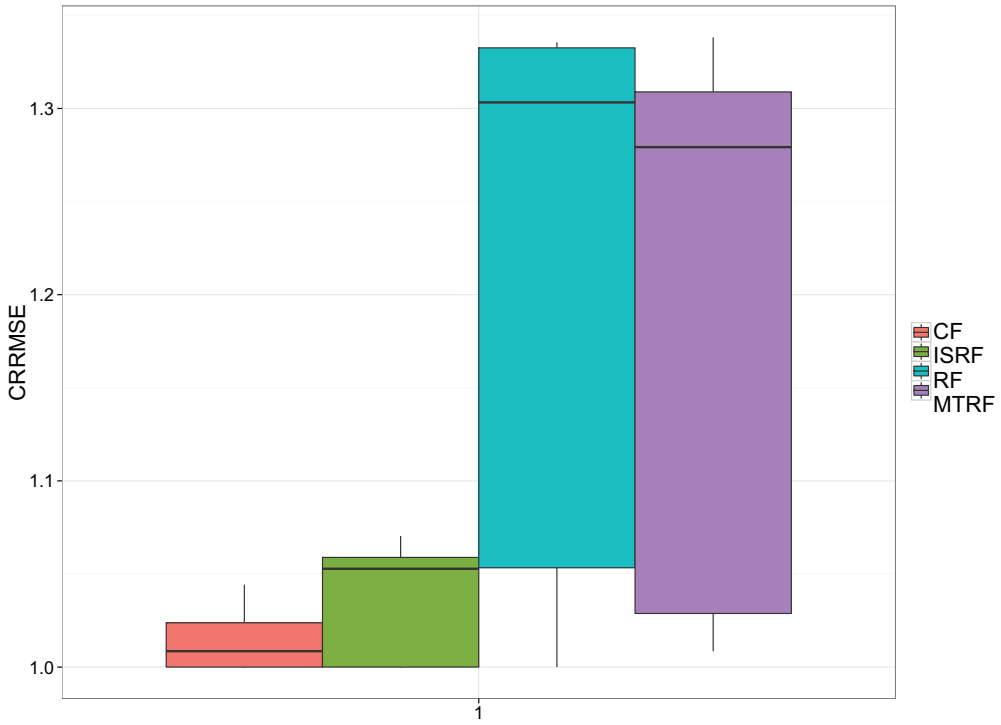
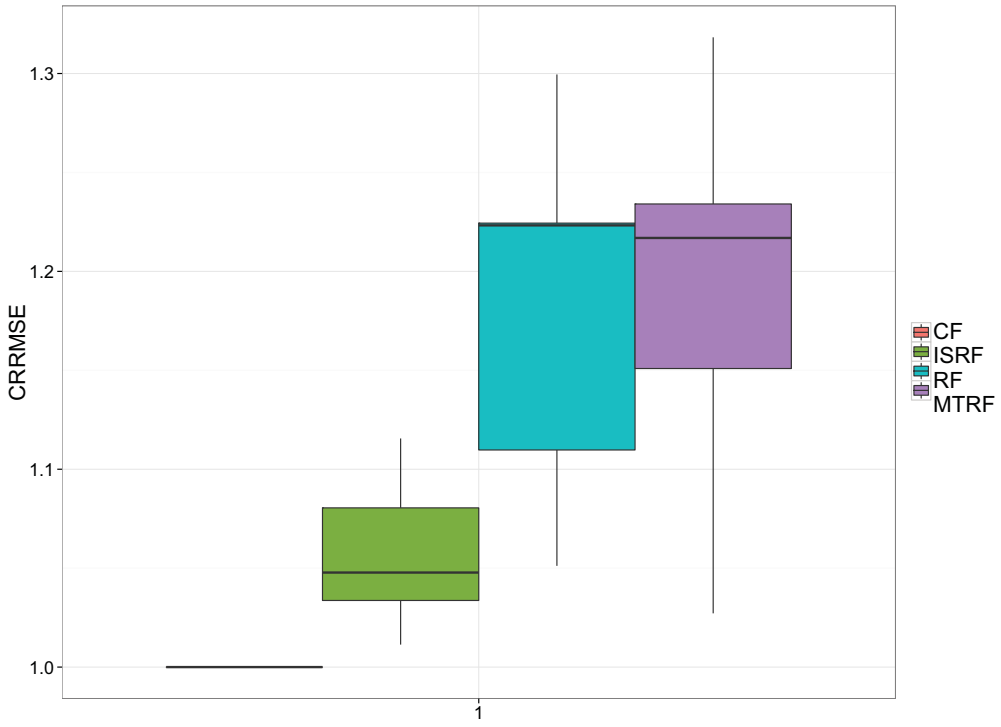


Figure 3.2: The CRRMSE of CF, ISRF, RF and MTRF for Synth 1 Data.

Synth 3

The third synthetic data consist of 40 input attributes and 20 unrelated target attributes ($\eta = 0.9$). Note that the increase in η induces lower relevancy amongst the target attributes. Figure 3.4 depicts the CRRMSE of the different methods over five replicates. As can be observed, CF continues to be strong competitor even with low relevancy across the large number of target attributes. This data set presents an example where there is no additional useful information to be shared between the target attributes which likely leads to a large number of non-relevant base learners. Nevertheless, the performance of CF is still comparable to the best method. The post-processing step makes CF robust even in the presence of a large number of non-relevant target attributes.

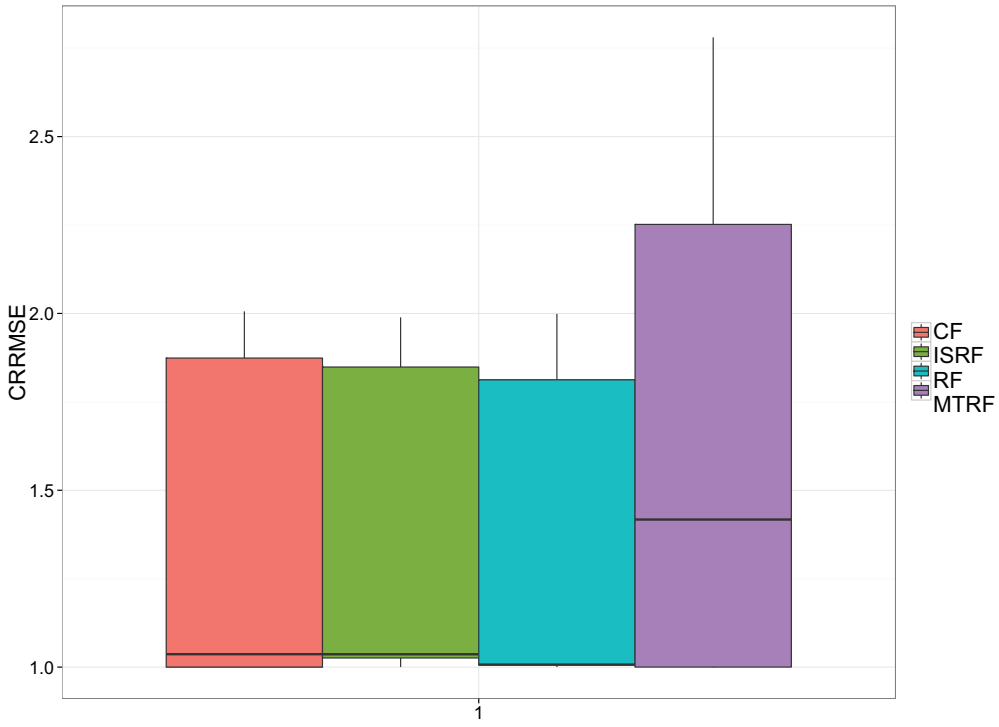
The experiments on the synthetic data sets allowed for the exploration of the



(a)

Figure 3.3: The CRRMSE of CF, ISRF, RF and MTRF for Synth 2 Data.

properties of CF under different properties of the data sets (number and relevancy of the target attributes). We conclude that CF improves prediction performance by leveraging useful information between related target attributes while remaining robust in the presence of non-related target attributes. The improvement of CF (relative to other methods) increases with larger number of target attributes that are more related to each other (smaller η values). In such cases, there are diverse information from the other target attributes that are useful for predicting τ. This in turn, allows for a construction of a highly diverse, yet strong, set of base learners.



(a)

Figure 3.4: The CRRMSE of CF, ISRF, RF and MTRF for Synth 3 Data.

3.4.2 Real Data

In this section we introduces a collection of real world multi-target problems from different domains such as robotics, marketing, education and process control. We describe each data set and discuss the results of our method on each one of them below.

SARCOS Data

This data relates to an inverse dynamics problem for a seven degrees-of-freedom SARCOS anthropomorphic robot arm. There are 21 input attributes on joint position, velocities and acceleration and seven attributes on joint torques. The data consists of 48933 instances and is available at http://www.gaussianprocess.org/. In this work,

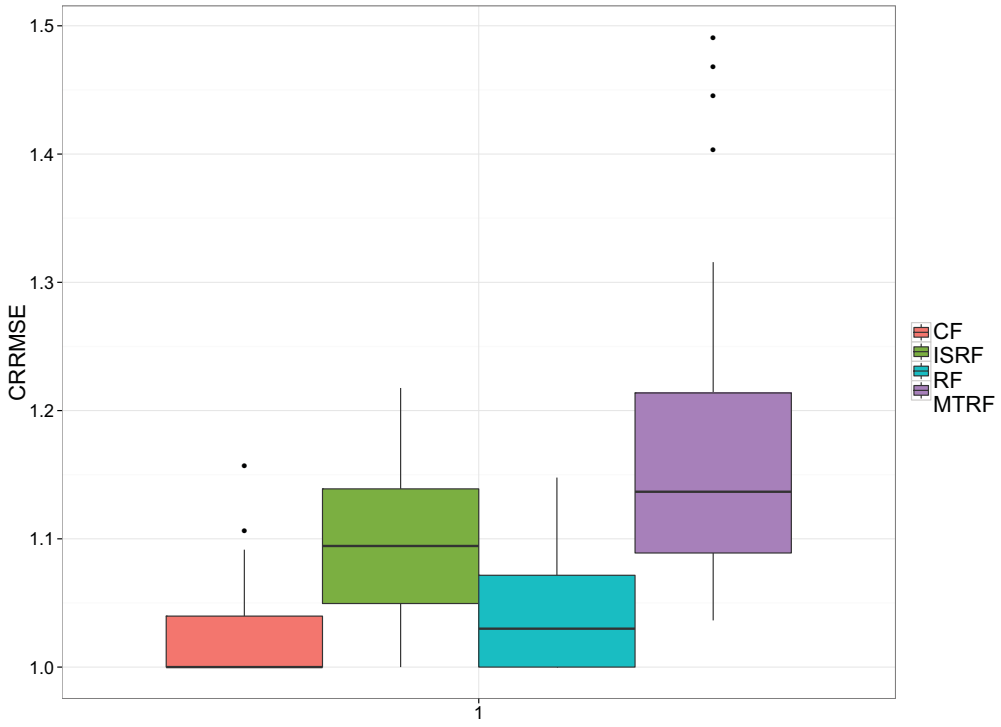
the attributes on joint torques are used as seven target attributes. Table 3.3 shows the average RRMSE of the seven target attributes over five replicates. The lowest RRMSE is shown in bold. CF outperforms the other methods in five of the seven target attributes. Figure 3.5 depicts the excellent CRRMSE performance of CF.

Target attribute	CF	ISRF	RF	MTRF
1	**0.411**	0.428	0.457	0.585
2	**0.407**	0.437	0.429	0.488
3	**0.399**	0.455	0.413	0.436
4	**0.352**	0.388	0.358	0.392
5	0.472	0.484	**0.448**	0.476
6	0.470	0.494	**0.447**	0.501
7	**0.318**	0.348	0.339	0.366

Table 3.3: Average RRMSE of the Seven Target Attributes for the SARCOS Data.

Customer Satisfaction (CS) Data

This data consists of 11 attributes on price and quality of service and 3 attributes on customer satisfaction with 400 instances (Esposito Vinzi *et al.*, 2007). The attributes on customer satisfaction are used as three target attributes. Table 3.4 shows the average RRMSE of the three target attributes over five replicates. The results show that CF outperforms the competitors in all target attributes (shown in bold). Figure 3.6 depicts the CRRMSE averaged across the three target attributes which depicts CF's good performance.



(a)

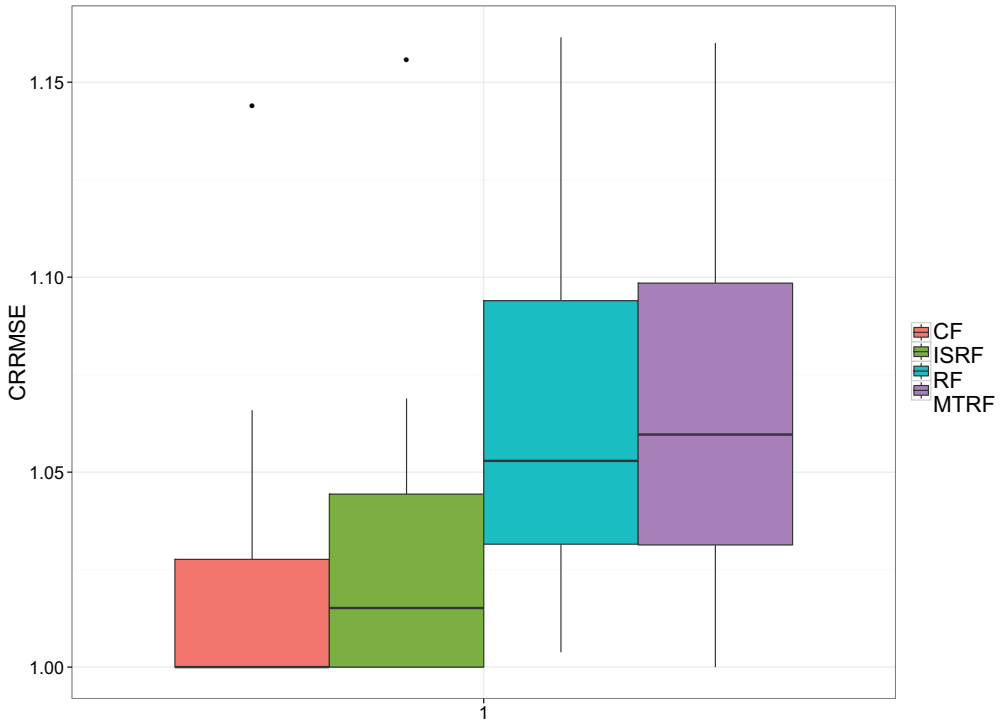
Figure 3.5: The CRRMSE of CF, ISRF, RF and MTRF for the Seven Target Attributes of SARCOS Data.

Target attribute	CF	ISRF	RF	MTRF
1	**0.356**	0.362	0.374	0.377
2	**0.363**	0.364	0.373	0.374
3	**0.373**	0.373	0.389	0.385

Table 3.4: Average RRMSE of the Three Target Attributes for the Customer Satisfaction Data.

Berkeley Guidance (BG) Data

This data consists of physical measurements 136 children born in 1928-29 in Berkley, CA during childhood (seven attributes) and adolescence (four attributes)(Tuddenham and Snyder, 1953). The attributes on adolescence physical measurements are used



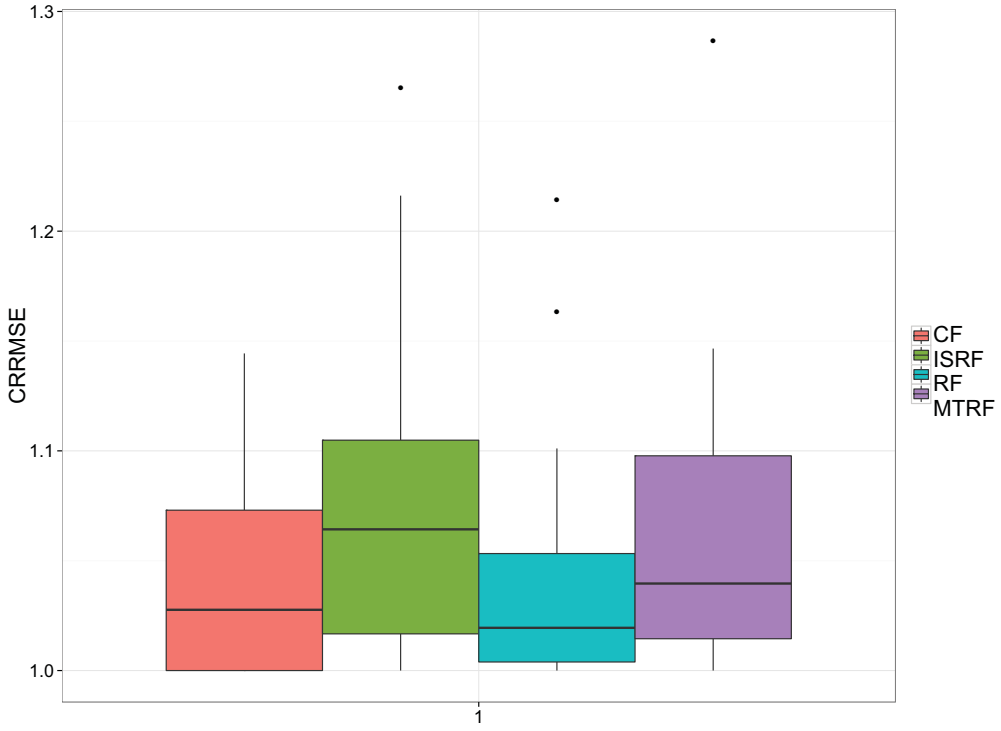
(a)

Figure 3.6: The CRRMSE of CF, ISRF, RF and MTRF for the Three Target Attributes of Customer Satisfaction Data.

as four target attributes. Table 3.5 shows the average RRMSE of the four target attributes over five replicates. The lowest RRMSE is shown in bold. CF outperforms the other methods in two of the four target attributes. Figure 3.7 depicts the CRRMSE averaged across the target attributes that conveys RF and CF are close competitors for this data set.

Target attribute	CF	ISRF	RF	MT
1	**0.726**	0.778	0.738	0.765
2	0.633	0.659	**0.625**	0.637
3	**0.762**	0.772	0.782	0.780
4	0.687	0.693	**0.677**	0.691

Table 3.5: Average RRMSE of the Four Target Attributes for the Berkeley Guidance Data.



(a)

Figure 3.7: The CRRMSE of CF, ISRF, RF and MTRF for the Four Target Attributes Of the Berkeley Guidance Data.

High School and Beyond (HSB) Data

Data collected from high school and secondary school students with 14 attributes and 600 instances. The data consists of 9 attributes on demographics, motivation

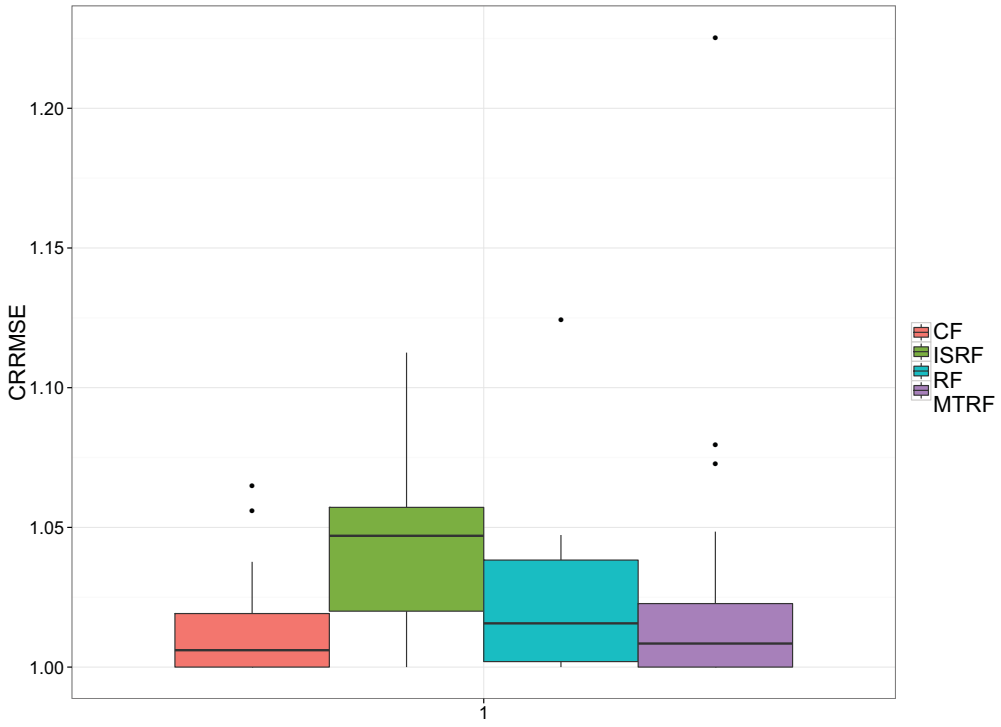
and school of the students, as well as five attributes on the students' standardized exam scores. Details are provided in Tatsuoka and Lohnes (1988). We use the exam scores as five target attributes. Table 3.6 shows the average RRMSE of the five target attributes over five replicates. The lowest RRMSE is shown in bold. CF outperforms the other methods in three of the five target attributes. Figure 3.8 depicts the CRRMSE averaged across the five target attributes which depicts CF's good performance for this data set.

Target attribute	CF	ISRF	RF	MTRF
1	**0.841**	0.872	0.844	0.851
2	0.823	0.857	0.837	**0.814**
3	**0.830**	0.857	0.844	0.860
4	0.847	0.868	0.843	**0.837**
5	**0.897**	0.908	0.903	0.920

Table 3.6: Average RRMSE Of the Four Target Attributes for the High School and Beyond Data.

Low-Density Polyethylene Production Process (LDP) Data

This is data from a low-density polyethylene production process. There are 14 process attributes and five quality attributes with 50 instances. More details of the data can be found in MacGregor *et al.* (1994). For our purpose, the five quality attributes are used as the target attributes. Table 3.7 shows the average RRMSE of the five target attributes over five replicates. The lowest RRMSE is shown in bold. CF outperforms the other methods in three of the five target attributes. Figure 3.9 depicts the CRRMSE averaged across the three target attributes that provides further evidence for CF's excellent performance.

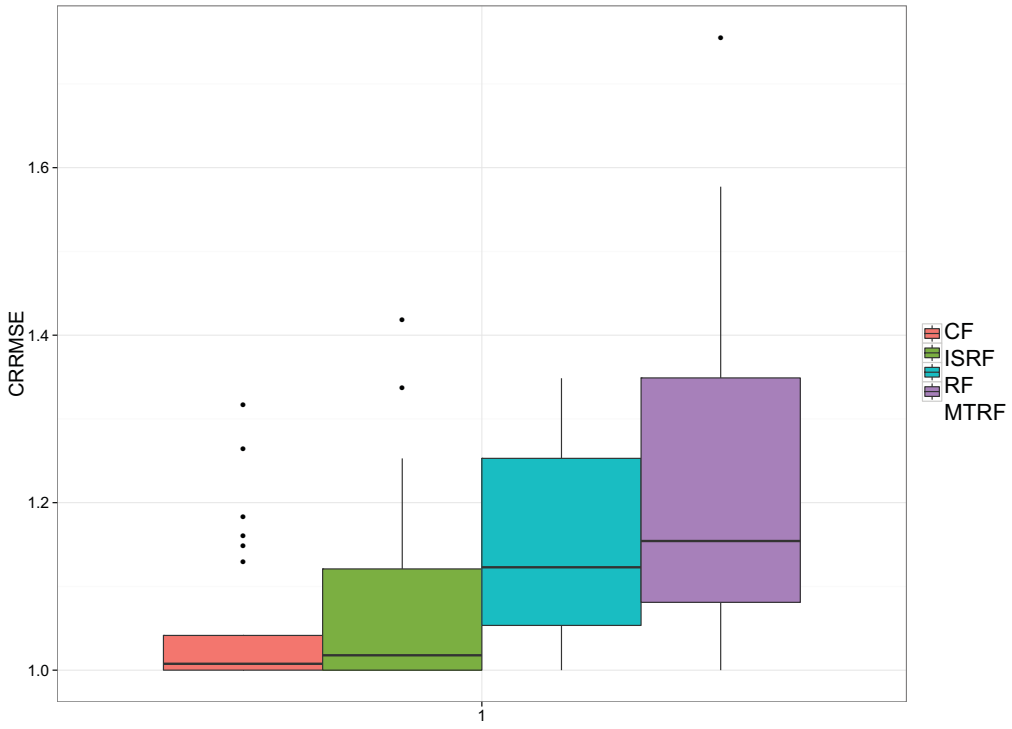


(a)

Figure 3.8: The CRRMSE of CF, ISRF, RF and MTRF for the Four Target Attributes of the High School and Beyond Data.

Target attribute	CF	ISRF	RF	MTRF
1	**0.574**	0.589	0.607	0.625
2	0.402	**0.395**	0.439	0.455
3	**0.785**	0.825	0.807	0.817
4	0.487	**0.484**	0.508	0.550
5	**0.360**	0.375	0.427	0.487

Table 3.7: Average RRMSE Of the Four Target Attributes for the Low-Density Polyethylene Production Process Data.



(a)

Figure 3.9: The CRRMSE Of CF, ISRF, RF and MTRF for the Four Target Attributes of the Low-Density Polyethylene Production Process Data.

3.4.3 Statistical Comparison

The one-sided Wilcoxon signed-ranks test is performed for pairwise comparison of CF and each of the competitors. The presence of multiple target attributes allows for two different approaches. The first treats each target attribute as an independent measure while the second computes the average over all target attributes in each data set and considers each average as an independent measure. These two approaches are used in Aho *et al.* (2012). Table 3.8 summarizes the p-values that reflect the significance of the CF's performance improvement.

Overall, the results of the experiments provide significant evidence for the benefit of CF, with the biggest improvements resulting from training the base learners

on a large number of relevant target attributes. We show the versatility of CF in handling these characteristics in real data from different domains. Furthermore, the experiments depict the robustness of the method in the presence of a large number of target attributes that are of low relevance to each other. All of these attest to the superiority of CF in handling multi-target prediction.

Furthermore, as mentioned ISRF and RF consist of $T \times q$ trees per target attribute, resulting in a total of $T^2 \times q$ trees for each data set. The CF and MTRF, on the other hand, consist of $T \times q$ trees that are shared across all target attributes. This sharing of the ensemble reduces the computations when the final goal is the prediction of more than a single target attribute.

Comparison	Over Averaged Target Attributes	Over Individual Target Attributes
CF vs ISRF	3.91E-03	3.20E-07
CF vs RF	2.73E-02	6.92E-04
CF vs MTRF	3.91E-03	2.46E-07

Table 3.8: P-values for Pairwise Comparison of CF to Each of the Competitors Using the One-Sided Wilcoxon's Test ($H_1 : RRMSE^{CF} < RRMSE^d$).

3.4.4 Computational Time

We next study the empirical computational time for the CF method. The SARCOS data set is used due its large number of instances and target attributes which allow us to construct data sets with different number of training instances and target attributes. We consider the training time for constructing $T \times 50$ trees in CF and performing the SGL weight assignment. It should be noted that results are shown for optimized code that predict the same target attribute that was used in training the tree but the same results should apply to cases where different target attributes are predicted. The scaled time is reported in Figure 3.10. The loglinear time is also

depicted by the black dotted line. As discussed in Section 3.3.4, the complexity of the CF method is loglinear which is validated by empirical results of Figure 3.10.

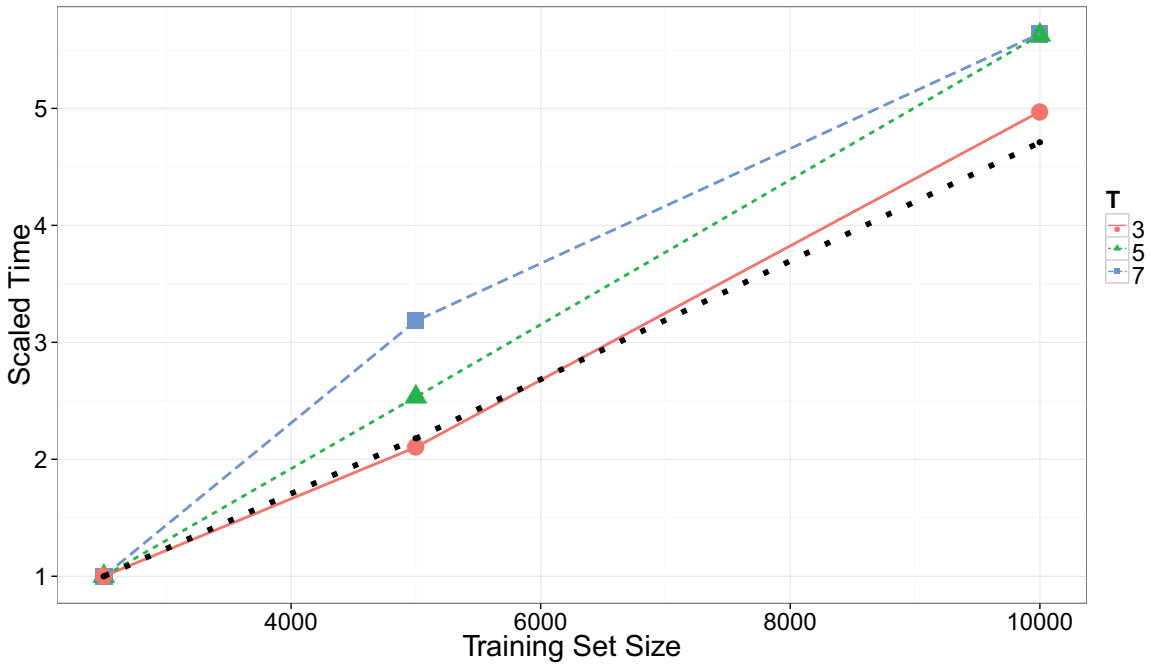


Figure 3.10: Scaled Training Time for Constructing $T \times 50$ Trees in CF and Performing the SGL Weight Assignment for Data Sets with Different Training Set Sizes and Different Number of Target Attributes. The Loglinear Time is Also Depicted by the Dotted Black Line.

3.5 Conclusions

Many real world problems involve the prediction of several target attributes. A new tree ensemble model called a compound forest (CF) is proposed, which exploits the different target attributes in forming a collection of diverse, yet strong, base learners. The weight assignment of the base learners in the final prediction is obtained through solving a regularized regression problem that takes into account the target attribute used for base learner training and the relevance of each learner for the prediction. The performance of the method is evaluated on synthetic and real data illustrating the benefits of the method.

In the current implementation of CF, the assignment of weights is assigned through solving a regularized regression problem that takes other base learners into account. However, since the base learners are constructed in a parallel fashion, they are constructed independently without taking into account the other base learners. Future work can include the development of a serial approach that takes the previous constructed base learners into account. Another interesting direction is clustering over the target attributes to group similar target attributes together (Thrun and O'Sullivan, 1996). The distance measure for this clustering may be a function of a node impurity of the partitions of the feature space obtained using one target attribute with respect to the other target attributes.

Chapter 4

MONITORING TEMPORAL HOMOGENEITY IN NETWORK STREAMS WITH A LIKELIHOOD RATIO TEST

4.1 Introduction

Statistical process control has widely been used towards monitoring various types of systems. We focus on monitoring complex systems that are modeled as networks. Such models can represent the complexities of many real world systems such as social, cyber and biological systems. The dynamics of entity relationships in such systems are important and need to be captured and monitored through network models. For example, the dynamics of email logs over time is better modeled through a stream of network snapshots at discrete time stamps compared to a single static network. We focus on monitoring such network streams for the quick detection of temporal behavior change through statistical monitoring. The objective is to learn the *reasons* behind network edge formation during a reference time period, characterizing typical system conditions, and to quickly detect time periods when edges are formed due to fundamentally different reasons. In other words, we are interested in testing *temporal homogeneity* in the network stream. In a social network, for example, it may be of interest to detect time periods that exhibit aberrations in friend formation. The start of the academic year that prompts friend formation within users of the same major may mark such an aberration. Note that this is different to testing for *static homogeneity* that aims to detect networks that have anomalous edges with respect to the rest of the current network.

Considerable amount of research has been devoted to modeling entity relationships

through static networks. Such approaches model entity relationships at a single time stamp or an aggregate view of the relationships over time through a single static network. The simplest of these is the Erdos-Renyi random graph model that describes networks where edges are formed independently between each pair of vertices with a common probability (Erdos and Renyi, 1959). This is an overly simple model and various attempts have been made to model systematic deviations from pure randomness (Frank and Strauss, 1986; Hoff *et al.*, 2002; Wang and Wong, 1987). As an example, the stochastic blockmodels (SBM) (Wang and Wong, 1987) is a multi-class extension of the Erdos-Renyi model. This model assigns a class to each vertex and uses a different edge probability for each pair of classes.

The underlying systems that are modeled through networks usually possess temporal dynamics. For example, in a social network, edges (friendship ties) may be added or deleted through time. The static models, mentioned in the previous paragraph, fail to model the underlying temporal dynamics (i.e. change in the topology of network through time). Incorporating the temporal aspect, previous work has focused on modeling the growth of networks. The simplest of these is to view the Erdos-Renyi random graph model as a dynamic network that starts with the unconnected set of vertices and adds a different edge to the network with fixed probability at each subsequent time stamp. Other work in this direction include Barabási and Albert (1999), Leskovec *et al.* (2007), Chakrabarti *et al.* (2004) and Pennock *et al.* (2002). In parallel, other work model the evolution of networks where vertices and edges are both created and deleted over time Hanneke *et al.* (2010); Ho *et al.* (2011); Sarkar and Moore (2005); Snijders (2005); Xu and Hero III (2013).

More recently, focus has been drawn on network monitoring for anomaly detection. Such efforts are usually tailored around two objectives that we refer to as testing for static homogeneity and testing for temporal homogeneity. Testing for static ho-

mogeneity aims to detect networks that have anomalous edges with respect to the current network (see for example Miller *et al.* (2013)). Testing for temporal homogeneity, on the other hand, aims to detect networks that have anomalous edges with respect to edges in the past networks. This is an important problem as changes in the system are likely reflected in the network and is the focus of the present and next chapter.

A typical approach towards testing temporal homogeneity is to monitor extracted measures from the network topology through time. As an example, McCulloh and Carley (2011) constructed control charts over different network measures. The work by Priebe *et al.* (2005), Marchette (2012) and Neil *et al.* (2014) monitored scan statistics for this purpose. Similarly, the work by Park *et al.* (2013) used a fusion of network statistics (including the scan static) to detect changes in a stream of networks.

The cited work are based on monitoring extracted measures from the network topology which can restrict their application to detecting only specific forms of anomaly in the network. For example, monitoring some measures are appropriate for detecting overall changes on the entire network, while others, are appropriate only for detecting changes in specific, defined, windows on the network (anomalies over paths and stars for example). In addition to the topological structure of the network, many real networks are augmented with vertex attributes which can be used towards a more general monitoring approach. For example, a social network is composed of friendship ties as well some attributes such as gender, age, etc. An academic citation network constitutes paper citations but also contains attributes on the papers such as the research interests and sum of published papers of the authors. Biological networks entail connectivity information but also include genes or protein characteristics of the vertices. This chapter presents a new method that models and monitors

the underlying network formation mechanisms via the vertex attributes through time and detects anomalies when this mechanism is under change. This mechanism assigns probabilities to the existence of each possible edge and, therefore, gives rise to the network. Our approach leverages vertex attributes in modeling and monitoring this mechanism through a logistic regression framework. The next section elaborates the motivation of the method followed with a detailed explanation in Section 4.3. Section 4.4 presents two case studies including monitoring Enron's dynamic email network and Section 4.5 provides experiments on simulated dynamic networks. Finally Section 4.6 gives some concluding remarks and direction for future work.

4.2 Background and Motivation

This section presents the motivation behind the proposed approach. We start with a short review on some network measures extracted from the network topological structure and follow with the application of these measures to detect different temporal inhomogeneities. The shortcomings of approaches based on these measures are illustrated, motivating the monitoring of the underlying network formation mechanism via the attributes.

Many different network measures have been developed through the years (Freeman, 1979, 1977; Wasserman, 1994). These measures are generally extracted at both vertex and network level. The relative importance of a vertex within a network is captured through vertex level measures such as degree, closeness and betweenness. The degree of a vertex is simply the number of its adjacent edges, closeness is the number of edges needed to access every other vertex and betweenness is the number of geodesics (shortest paths) going through the vertex. It should be noted that such vertex level measures are sometimes averaged across the network to provide an overall measure for the whole network. Other network measures are captured at the network

level and reflect the structure of the overall network. As an example, network density refers to the ratio of the number of edges and the number of possible edges.

Additional network measures may be captured through scan statics (Marchette, 2012; Neil *et al.*, 2014; Park *et al.*, 2013; Priebe *et al.*, 2005). The construction of the scan statistic involves enumerating fixed, defined, windows over the entire network. For example, Priebe *et al.* (2005) considered the window as kth-order neighborhoods around each vertex. This neighborhood is defined as the subnetwork composed of the vertices that have a geodesics of length k or smaller to the vertex. A locality statistic, such as the number of edges in this neighborhood, is then calculated and a function, such as the maximum, of the locality statistic over all vertices is taken to be the scan statistic. Similarly Neil *et al.* (2014) enumerated star and paths over the network for the construction of the scan statistic.

We start by considering the network in Part (a) of Figure 4.1 where vertices are connected homogeneously (statically homogeneous). An external event results in excessive communication over the entire network resulting in the network depicted in Part (b). Such a change in the network is reflected in the network measures (for example, degree) and thus allow for detection by their monitoring through these measures.

A more interesting temporal inhomogeneity is what is known as the "chatter" anomaly (Park *et al.*, 2013). Here, a small unspecified subset of the vertices have excessive communication during some time period. An example is shown in Part (c) of Figure 4.1 where local excessive communication in observed. An approach for the detection of such non-homogeneities is to monitor partitions of the network. This is, however, challenging given the absence of prior knowledge about the location of non-homogeneities. Previous works Marchette (2012); Neil *et al.* (2014); Park *et al.* (2013); Priebe *et al.* (2005) enumerated fixed, defined, windows over the entire network to

construct a scan statistic. For example, as mentioned, the work in Priebe *et al.* (2005) enumerates k-th order neighborhoods on the network while Neil *et al.* (2014) enumerates star and path over the network. Relying solely on the network's topology, such approaches resort to an exhaustive search over the entire network based on defined windows. A shortcoming of such approaches is the restricted search performed on the defined windows making such approaches appropriate only for identifying specific shapes of anomalies (anomalies over paths and stars or k-th order neighborhoods for example).

Besides the network topology, many networks include vertex attributes that may be useful for the identification of the non-homogeneous region. The networks in Figure 4.1 Parts (b) and (c) are revisited in Figure 4.2 by incorporating such attributes (each vertex is associated with a unique ID and two attributes are shown in color and size). These figures shed light on the location of non-homogeneity through the attributes: namely, that the excessive communication is amongst vertices of the same color. Note that this change is more precisely described as excessive activity in local regions of the attribute space and is, thus, better detected through a monitoring approach that leverages the attributes.

These examples of change present cases where the underlying mechanism behind edge formation is under change. Considering the vertex attributes, each edge is placed in an attribute space and the underlying network-formation mechanisms assigns probabilities for the existence of each edge according to its location in this space. This in turn derives the observed network. The mechanism generating the network in Part (a) of Figure 4.1 assigns equal probability p_0 to each edge regardless of its location in the attribute space. This mechanism changes and gives rise to the network observed in Parts (b) and (c). Part (b) presents a case where the mechanism assigns equal probability p_1 ($p_0 < p_1$) to each edge (again, regardless of the edge's location

in the attribute space) and Part (c) presents a case where higher probabilities are assigned to edges that lie in a local region of the attribute space (assign $p_2 > p_0$ for all edges that have same colored vertex and p_0 otherwise). This motivates a general approach for testing temporal homogeneity in network steams that directly monitors the underlying mechanism that forms the network. Our approach integrates attributes in network monitoring extending previous work that have integrated such attributes in other network modeling tasks such as link prediction and attribute inference (Al Hasan *et al.*, 2006; Gong *et al.*, 2011; Kim and Leskovec, 2010; Kumar *et al.*, 2004).

As a final example, consider the email network of employees of the Enron corpus throughout the course of its history (Priebe *et al.*, 2005). Each employee is represented as a vertex and weekly email communication is aggregated to form network edges at weekly time stamps. Furthermore, each employee is associated with a role in the company and the probability of an email communication between two employees may be modeled as a function of the pair's role combination. The work of Xu and Hero III (2013), with the objective of network evolution modeling, demonstrates that key events in Enron's history are reflected through changes of employee email communication. For example, a CEO's resignation results in an increase (compared to past) in email communication of other CEOs. By monitoring the edge formation mechanism, the proposed method leverages the roles of the employees for the identification of the temporal change (the local temporal inhomogeneity of increased CEO communication). We will return to this example in more detail later in the chapter.

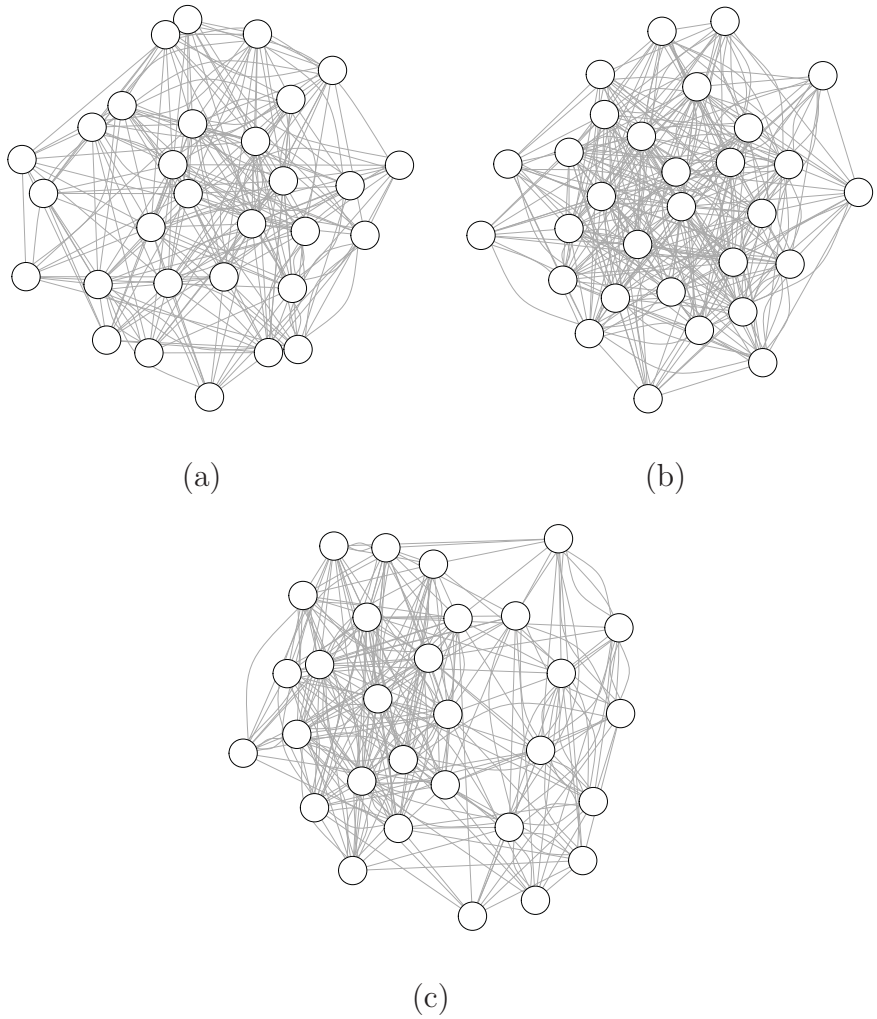
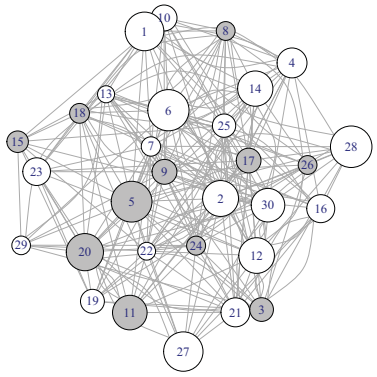


Figure 4.1: Example of Some Changes in Networks.

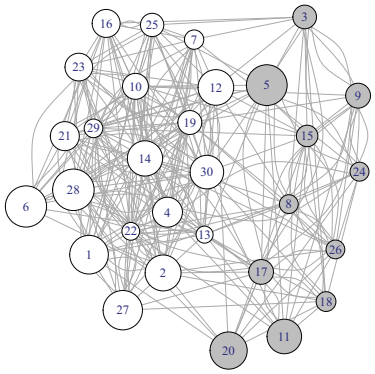
4.3 Monitoring Network Formation Mechanisms

4.3.1 Method

This section presents the details of the proposed method for monitoring the network formation mechanism. The objective is to monitor a stream of networks $G(t) = (V(t), Y(t))$, $t = 1, \cdots$, characterized by the vertices $V(t)$ and edges $Y(t)$. Note that each discrete time stamp provides a separate network. Without loss of generality, we focus on network streams with fixed vertices such that $V(t) = V =$



(a)



(b)

Figure 4.2: An Example of Excess Activity in Local Regions of the Attribute Space.

$\{v_1, \cdots, v_i, \cdots, v_\nu\}$ and $Y(t) = \{y_{1\,2}(t), \cdots, y_{i\,j}(t), \cdots, y_{\nu-1\,\nu}(t)\}$ where v_i denotes vertex i, $y_{ij}(t)$ denotes the edge between v_i and v_j at time t, ν is the number of vertices and η the number of all possible edges. Each edge $y_{ij}(t)$ takes on two possible values of 0, indicating its absence, and 1 indicating its presence, at time stamp t. At each time stamp, $y_{ij}(t)$ is modeled through a vector of P attributes $x_{ij} = (x_{1ij}(t), \cdots, x_{Pij}(t))$. Although, the definition of these attributes is problem-specific, some general guidelines hold. For example, Al Hasan *et al.* (2006) discussed attributes that represent proximity between the pair of vertices and attributes that are aggregation of the pair's attributes. As an example, consider an email network data in an university. Here,

the age difference of the two users, whether they share the same major, the sum of the number of classes they are enrolled in may be influential on the email exchanges and be considered as the associated attributes.

Once the attributes are determined, a method is needed to model the probability of communication between v_i and v_j, denoted by $\theta_{ij}(t)$, as a function of the attributes at time t. We regard this model (and its parameters) as the mechanism that is generating the network and look for changes in this mechanism through time. We assume each edge is a Bernoulli random variable and model the log-odds of $\theta_{ij}(t)$ as a linear function of the attributes. We adopt the logistic regression model (Myers *et al.*, 2012) where $y_{ij}(t) \sim \text{Bernoulli}(\theta_{ij}(t))$ and

$$\theta_{ij}(t) = P(y_{ij}(t) = 1 \mid x_{ij}(t)) = \text{logit}^{-1}(\sum_{p=1}^{P} \beta_p \, x_{pij}(t)) \tag{4.1}$$

Similar Bernoulli models have been adopted to model edges in networks. For example, the work in Perry *et al.* (2013) uses a Bernoulli model for network edges in the context of cluster detection and Miller *et al.* (2013) for testing static homogeneity.

Assume a reference network set of size q, denoted by $\{G(t), t \in R\}$, where R denotes the set of the corresponding time indices, is available. This reference set is collected during typical conditions of the system under study. Testing for temporal homogeneity is achieved by comparing the current incoming network to this set. Hence, upon receiving $G(\tau), \tau = 1, \cdots, T$, we test if the mechanism behind its formation is the same as the networks in the reference set. Assume that a different mechanism has indeed generated $G(\tau)$ (in comparison to reference set). Then, under the logistic regression model

$$\theta_{ij}(t) = \begin{cases} \text{logit}^{-1}(\sum_{p=1}^{P} {\beta_p}^0 \, x_{pij}(t)) & \text{for } t \in R \\ \text{logit}^{-1}(\sum_{p=1}^{P} {\beta_p}^1 \, x_{pij}(t)) & \text{for } t = \tau \end{cases}$$

where $\beta^0 = (\beta_1^0, \cdots, \beta_P^0)$ denotes the vector of shared parameter for the reference set and $\beta^1 = (\beta_1^1, \cdots, \beta_P^1)$ denotes the vector of coefficients after change. To check if the change has occurred, we need to test

$$H_0 : \beta^1 = \beta^0$$
$$H_1 : \beta^1 \neq \beta^0 \tag{4.2}$$

We consider a likelihood ratio test (LRT) to test this hypothesis. Methods based on LRT for change detection have been successfully applied to different problem domains (refer to Paynabar *et al.* (2012); Sullivan and Woodall (1996) for examples). Denoting the Bernoulli probability mass function using $h(.)$, the log-likelihood function under the alternative can be written as

$$l_1 = \log\left\{\prod_{t\in R}^{\cdot}\prod_{i=1}^{\nu}\prod_{j\neq i}^{\cdot} h(y_{ij}(t);\, \beta^0, x_{ij}(t)) \times \prod_{i=1}^{\nu}\prod_{j\neq i}^{\cdot} h(y_{ij}(\tau); \beta^1, x_{ij}(\tau))\right\}$$
$$= \sum_{t\in R}^{\cdot}\sum_{i=1}^{\nu}\sum_{i\neq j}^{\cdot}\left\{y_{ij}(t)\,\mathrm{logit}(\theta_{ij}^0) + \log(1-\theta_{ij}^0)\right\} + \sum_{i=1}^{\nu}\sum_{i\neq j}^{\cdot}\left\{y_{ij}(\tau)\mathrm{logit}(\theta_{ij}^1) + \log(1-\theta_{ij}^1)\right\}$$

where θ_{ij}^0 and θ_{ij}^1 denote the probability of communication between v_i and v_j at time t obtained by substituting β^0 and β^1 in Equation 4.1 respectively. Let $\hat{\beta}^U$ denote the maximum likelihood (ML) estimate of β by training the logistic regression model using $\{G(t), t \in U\}$. In case of a single element U, we use the element's index as the name of the set (i.e. $\hat{\beta}^\tau$ denotes the ML estimate of β by training the logistic regression model using $G(\tau)$). The ML estimate of θ_{ij}^0 and θ_{ij}^1 under the alternative hypothesis are obtained from substituting $\hat{\beta}^R$ and $\hat{\beta}^\tau$ in Equation 4.1. We will denote these estimates by $\hat{\theta}_{ij}^R$ and $\hat{\theta}_{ij}^\tau$.

Similarly, the log-likelihood function under the null of no change can be written as

$$l_0 = \log\left\{\prod_{t\in R}^{\cdot}\prod_{i=1}^{\nu}\prod_{j\neq i}^{\cdot} h(y_{ij}(t);\beta^0, x_{ij}(t)) \times \prod_{i=1}^{\nu}\prod_{j\neq i}^{\cdot} h(y_{ij}(\tau);\beta^0, x_{ij}(\tau))\right\}$$

$$= \sum_{t\in R}^{\cdot}\sum_{i=1}^{\nu}\sum_{j\neq i}^{\cdot}\left\{y_{ij}(t)\text{logit}(\theta^0_{ij}) + \log(1-\theta^0_{ij})\right\} + \sum_{i=1}^{\nu}\sum_{j\neq i}^{\cdot}\left\{y_{ij}(\tau)\text{logit}(\theta^0_{ij}) + \log(1-\theta^0_{ij})\right\}$$

The ML estimate of $\theta^0_{ij}(t)$ under the null is obtained by substituting $\hat{\beta}^{R'}$, $R' = \cup(R,\tau)$ in Equation 4.1 which we denote by $\hat{\theta}^{R'}$. Replacing the parameters with their estimates and simplifying, the negative of the log-likelihood ratio can be written as

$$l_1 - l_0 = \sum_{t\in R}^{\cdot}\sum_{i=1}^{\nu}\sum_{j\neq i}^{\cdot}\left\{y_{ij}(t)[\text{logit}(\hat{\theta}^R_{ij}) - \text{logit}(\hat{\theta}^{R'}_{ij})] + \log(\frac{1-\hat{\theta}^R_{ij}}{1-\hat{\theta}^{R'}_{ij}})\right\}$$
$$+\sum_{i=1}^{\nu}\sum_{j\neq i}^{\cdot}\left\{y_{ij}(\tau)[\text{logit}(\hat{\theta}^\tau_{ij}) - \text{logit}(\hat{\theta}^{R'}_{ij})] + \log(\frac{1-\hat{\theta}^\tau_{ij}}{1-\hat{\theta}^{R'}_{ij}})\right\}$$

The asymptotic distribution of the LRT statistic, $\Lambda(\tau) = 2(l_1 - l_0)$, under the null hypothesis is chi-square with degrees of freedom equal to difference in the number of parameters for the null and alternative model (Myers *et al.*, 2012). Upon receiving each network, this value is calculated and is plotted against time to monitor for changes.

4.3.2 Variations

Different variations may be applied for conducting this approach in practice. The approach discussed thus far is to consider a set of reference networks, referred to as the reference network set (indexed by set R in the above formulation), and then to compare each incoming network to this set. We will refer to this approach as the static reference approach (SRq, where q is the number of networks in this set).

In practice, time is needed to accumulate a reference network set which impedes immediate monitoring. This motivates a "self-starting" approach (Capizzi and Masarotto, 2010; Maboudou-Tchao and Hawkins, 2011). One way to address this is

to dynamically update a small initial reference network set. That is, at each time stamp, the incoming network is examined and is entered into the reference set upon absence of signal. We refer to this approach as the dynamic reference (DR) approach.

In some applications, the mechanism generating the network may experience slow evolution over time. We refer to this phenomena as *inherent dynamic variation* of the network. In an email network for example, a slow trend that promotes emails between users of similar age may manifest on the network through a slow evolution. The detection of the inherent dynamic variation may not be of interest to us when focus is merely on detecting abrupt changes. To address this, a final modification of the proposed method is to consider a sliding window of reference networks that is updated dynamically. This approach allows for capturing up to date, typical behavior of the system, thereby, allowing for better detection of abrupt changes in the presence of inherent dynamic variation. We will refer to this as the dynamic reference sliding window of size (DRWq, where q is the size of the sliding window).

4.4 Case Studies

This section presents two case studies to illustrate the details of the proposed method. We fist consider monitoring simulated dynamic networks imitating email communication networks in a company. We then, revisit Enron's dynamic email network alluded to in Section 4.2.

4.4.1 Simulated Dynamic Networks

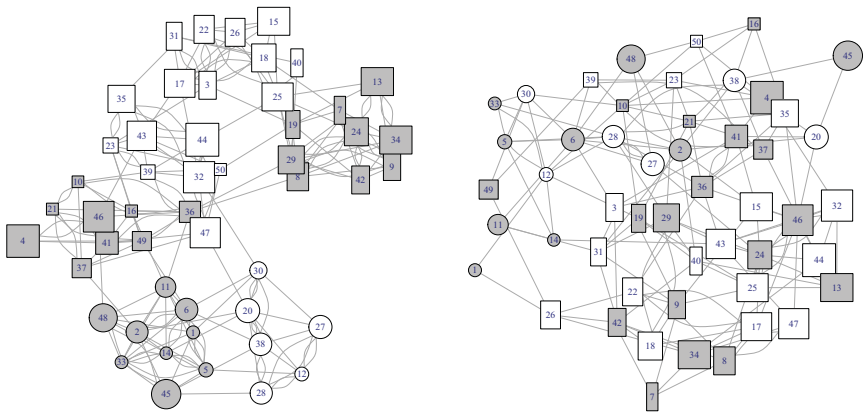
The simulated data imitates the email communication network of a company's team consisting of 50 members. The team members are distributed through two departments and differ with respect to rank (rank 1, 2, 3) and experience duration. All members work on a single project until completion before moving to the next project.

During a typical project, team members mainly work within their department, and with minimal inter department communication. Also, the hierarchy of the team derives higher communication between members of similar rank. The monthly email communication is modeled through a stream on networks. Part (a) of Figure 4.3 depicts the email communication between team members during a typical project (referred to as Project 0). Each member is represented as a vertex (with corresponding member ID) and an edge represents at least one email from v_i to v_j in month t. The vertices are shaped according to rank, sized according to experience and colored according to department. High connectivity within members of the same department and similar rank is apparent during Project 0.

Detecting An Abrupt Step Change

A new project (referred to as Project 1) demands interdisciplinary knowledge deriving inter departmental collaboration. Also, the project calls for some guidance from higher rank members to lower rank ones inducing communication between members of different rank. The communication network of the team during this project is depicted in Part (b) of Figure 4.3. To demonstrate the detection of a change, monthly email communication is monitored for a total of 100 Months. Month 25 marks the onset of Project 1. The reference set of 10 networks is collected from months in which the team works on typical projects that demand minimal inter departmental and rank communication. Figure 4.4 depicts the detection of this change by the monitoring $\Lambda(\tau)$ through the SR10 approach. We note that the results for DRW10 and DR are similar and are thus not shown.

Another project (Project 2) is considered that also requires inter departmental and rank communication. The inter departmental and rank communication, is however, much more subtle compared to Project 1. Part (b) of Figure 4.5 depicts the monthly



(a) (b)

Figure 4.3: Email Communication of a Team During Two Different Projects (Project 0 and Project 1).

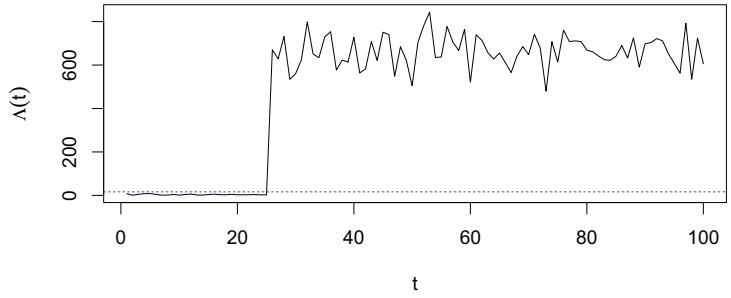
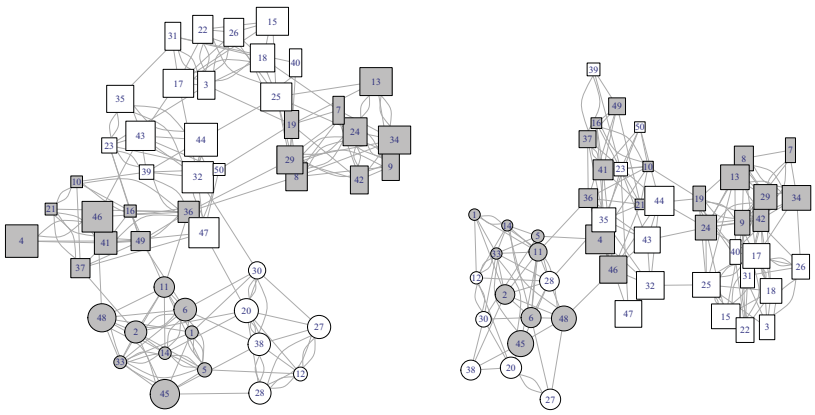


Figure 4.4: Plot of LRT Statistic Versus Time Using SR10. The Limit is Set to $\chi^2_{4,0.0027}$.

email communication of the team during this project. Part (a) depicts the email communication during Project 0, originally shown in Figure 4.3 and repeated for visual comparison. Figure 4.6 depicts the detection of this change by the monitoring $\Lambda(\tau)$ through the SR10 approach.



(a) (b)

Figure 4.5: Email Communication of a Team During Two Different Projects (Project 0 and Project 2).

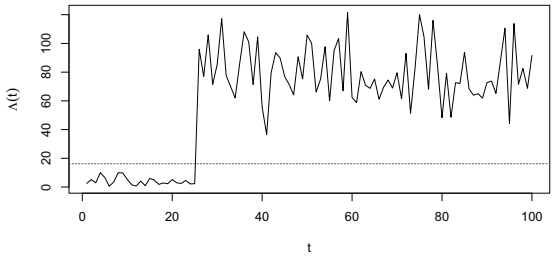


Figure 4.6: Plot of LRT Statistic Versus Time Using SR10.

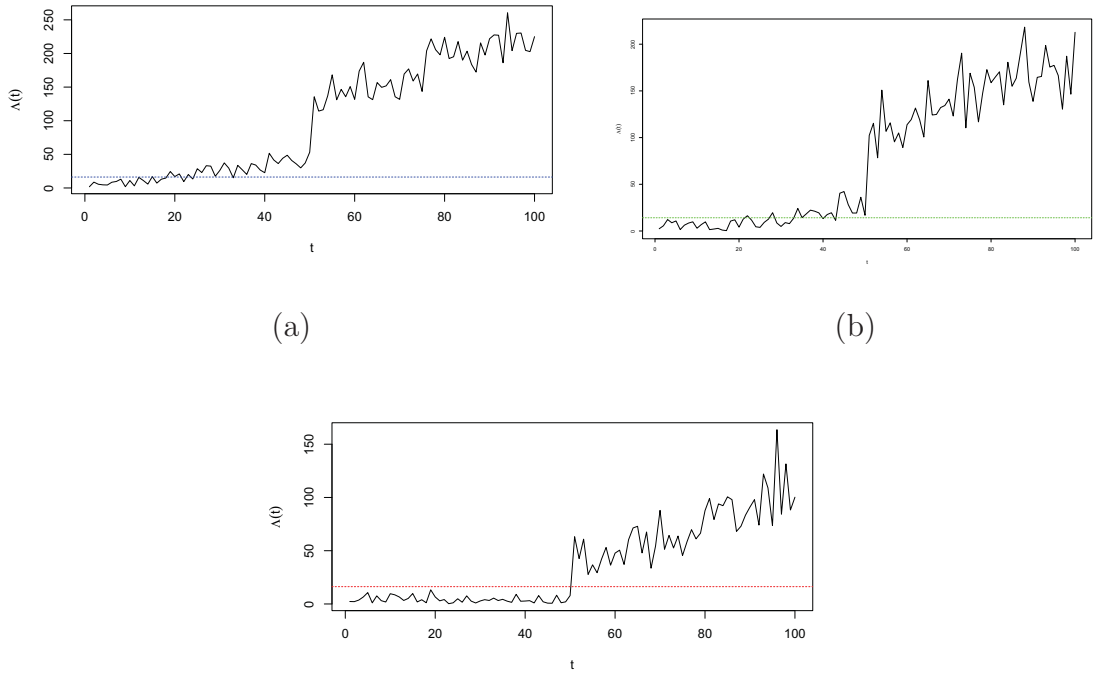
Detecting An Abrupt Change In The Presence Of Inherent Dynamic Variation

We now turn to the detection of a change in the team's communication in the presence of inherent dynamic variation. In this situation, higher communication between members who joined the team near the same time is observed over time. This slow inherent dynamic variation is irrespective of the project that the team works on. We are interested to detect time periods with abrupt aberrations in the team's intercommunication (such as when the team switches to work on a projects that demands atypical intercommunication).

To illustrate, we consider a new project (referred to as Project 3) that requires inter departmental and rank communication. The team initially works on a typical project (such as Project 0) and then switches to Project 3 at the 50th month. The mentioned inherent dynamic variation is present irrespective of the project the team works on. Figure 4.7 depicts the three proposed approaches for this detection. The results reflect on the differences of the approaches towards detecting a change in the presence of inherent dynamic variation. DRW10 is appropriate towards the objective of abrupt change detection in the presence of inherent dynamic variation. By dynamically updating the reference set through a sliding window, up to date, typical behavior is captured in the reference set that minimizes false alarm. The SR10 approach, on the other hand, generates a large number of false alarms as the reference data does not capture the up to date typical behavior. Ultimately, the choice of the approach depends on the objectives of monitoring : in the presence of inherent dynamic variation, DRW10 is appropriate for detecting an abrupt change, whereas SR10 is appropriate for detecting inherent dynamic variation.

4.4.2 Enron's Dynamic Email Network

This section demonstrates the application of the proposed method for monitoring a dynamic network from the Enron corpus (Priebe *et al.*, 2005). The data consists of email communications between Enron employees from 1998 to 2002. This is modeled as a stream of directed networks where an edge between two vertices indicates at least one email sent between the pair in a one week time interval. We take advantage of the recorded roles of the users to add attributes to the vertices. For simplicity, we restrict our attention to email communications between CEOs, directors and managers (pooled in to one category and referred to as DM) and presidents (PR). Therefore, each user under consideration has one of these roles. Somewhat similar



(a) (b)

(c)

Figure 4.7: Plots of the LRT Statistic Versus Time Using the Three Proposed Approaches.

to SBM, We use the role combinations of the pair as the attributes, resulting in a categorical attribute with nine possible values (CEO to CEO, CEO to DM, CEO to PR, etc). We use the SR4, DRW4 approach on this data, depicted in Figure 4.8. In the SR4 approach, each incoming network is compared to a static reference set composed of the four weekly networks in the first month of monitoring. As can be observed, many of the subsequent networks exhibit temporal inhomogeneity with respect to the networks in this reference set. This illustrates the high volatility of email communications over the monitoring years. The DRW4 approach, on the other hand, dynamically updates the reference set to include the networks over the most recent past month. The volatility of the email communication over the monitoring period (as observed in Part (a)) justifies the use of a small sized window. The dy-

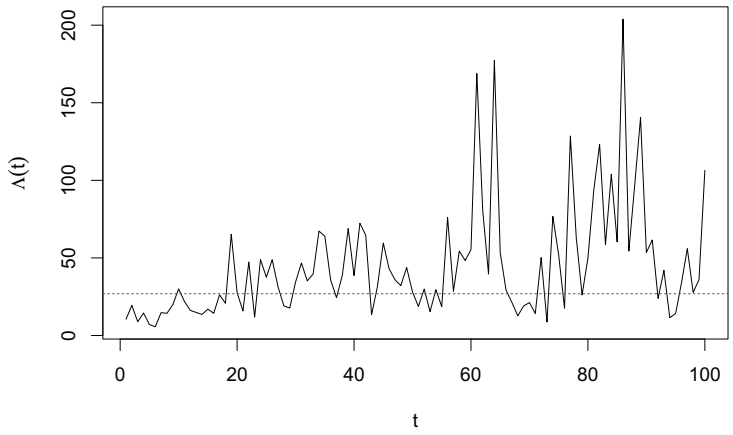
namic updating of the reference set allows the most recent behavior to be captured and, therefore, allows for comparing each incoming network to the networks in the most recent past month. Less temporal inhomogeneity is apparent in this approach which depicts short-time stationarity properties of the network stream. We notice three main spikes at around $t = 20, 60, 80$. Tracking the log of events in the Enron's scandal, these three time periods mark key events. The first of these marks the issue date of Enrons Code of Ethics. The second is around an extreme low point for Enrons stock and third around the time of Skilling's resignation.

The detected anomalies correspond to key events in the Enron scandal and are in line with the findings of other researchers such as (Priebe *et al.*, 2005; Xu and Hero III, 2013). Nevertheless, we examine our method further by assessing its ability to detect injected anomalies. Towards this end, we inject 20 additional emails amongst CEOs (CEO to CEO) in weeks 35 to 50. Figure 4.9 demonstrates the ability of the method to detect this excessive communication amongst the CEO's.

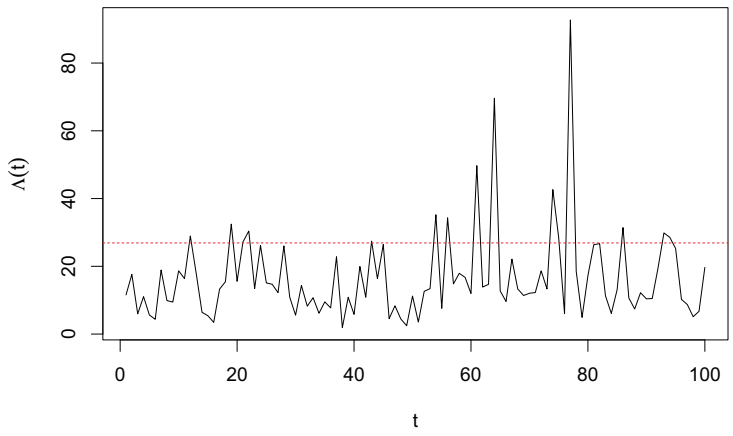
4.5 Experimental Evaluation

The performance of the proposed method is evaluated on simulated dynamic networks. We compare the performance of the SR approach with CUSUM charts based on network measures such as density, average degree, average closeness and average betweenness as proposed in McCulloh and Carley (2011). The CUSUM charts use standard parameter settings (the shift to be detected is set to 0.5 standard errors and the decision interval is set to 5 standard errors (Montgomery, 1991)).

As demonstrated in Section 4.4.1, the DR and DRW variations of the proposed method are more appropriate for detecting an abrupt change in the presence of inherent dynamic variation. On the contrary, the SR and CUSUM are appropriate for detecting small abrupt changes in the network and, thus, fail to detect an abrupt

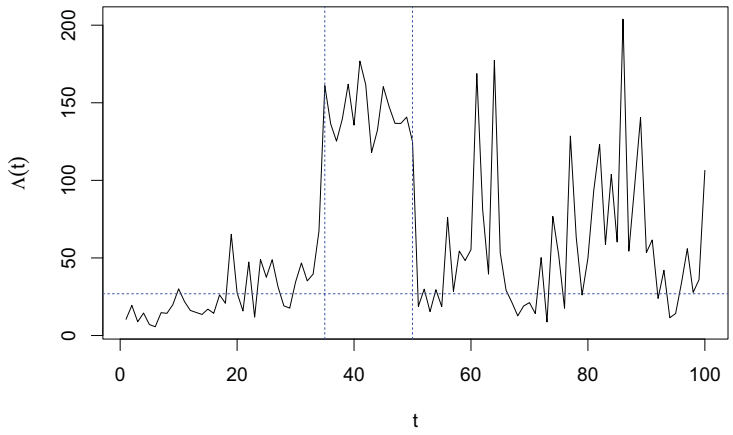


(a)

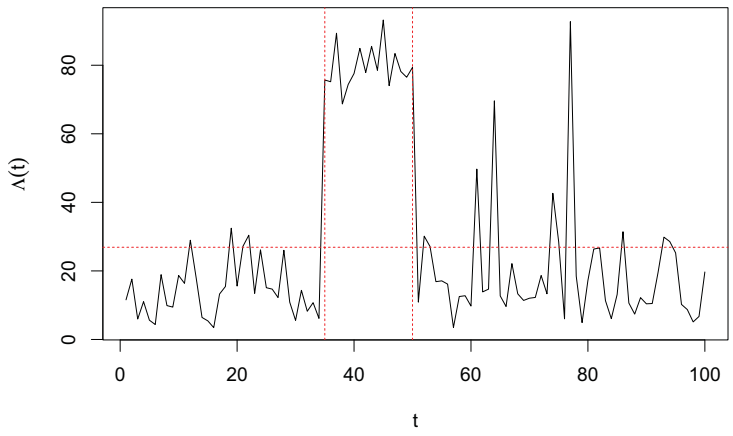


(b)

Figure 4.8: Plot of the LRT Static Versus time for Monitoring Weekly Emails of Enron's Employees using the SR4 and DRW4 Approach. The Control Limit is Set to $\chi^2_{9,0.0027}$.



(a)



(b)

Figure 4.9: Plot of the LRT Static Versus Time for Monitoring Weekly Emails of Enron's Employees in the Presence of Injected Change at $t = 35 - 50$ Using the SR4 and DRW4 Approaches (Parts (a), (b) Receptively).

change in the presence of inherent dynamic variation. The comparisons in this section, therefore, only includes the SR and CUSUM approaches as these two methods are designed for the same problem of detecting a small abrupt change.

Each simulated network consists of ν fixed vertices with two fixed attributes. The first attribute is $Uniform(20, 50)$ and is used to define X_1. The second attribute is $Bernoulli(0.5)$ and is used to define X_2. The reference set of 10 networks, also used in constructing the CUSUM charts, is generated according to

$$\theta_{ij} = \text{logit}^{-1}(0.1 - 0.2x_{1ij} + 0.3x_{2ij}) \tag{4.3}$$

Different changes are induced according to

$$\theta_{ij} = \text{logit}^{-1}(0.1 - 0.2x_{1ij} + (0.3 + \delta_1)x_{2ij} + \delta_2(1 - x_{2ij})) \tag{4.4}$$

We start by examining the run lengths (RL) of the different methods under no change. Figure 4.10 depicts the results where each box plot depicts the RLs of 100 streams each with 500 networks (The results are shown for networks for $\nu = 50$ vertices but our experiments indicate similar results for other values of ν). As depicted, the RL of the proposed method compares favorably to the other approaches. We note here that the control limit in the LRT approach is set to $\chi^2_{3,0.0027}$ for all experiments in this section.

Next, an experiment with networks with two different numbers of vertices ($\nu = 50, 100$) and different changes (δ_1 and δ_2) is conducted. The induced changes are summarized in Table 4.1.

A total of 100 different streams of length 50 are generated for each case. The first network that is detected to be an anomaly is considered as the run length of the procedure. In our evaluation, a change not detected for the entire duration of monitoring (a stream of 50 networks) is declared as undetected and the truncated

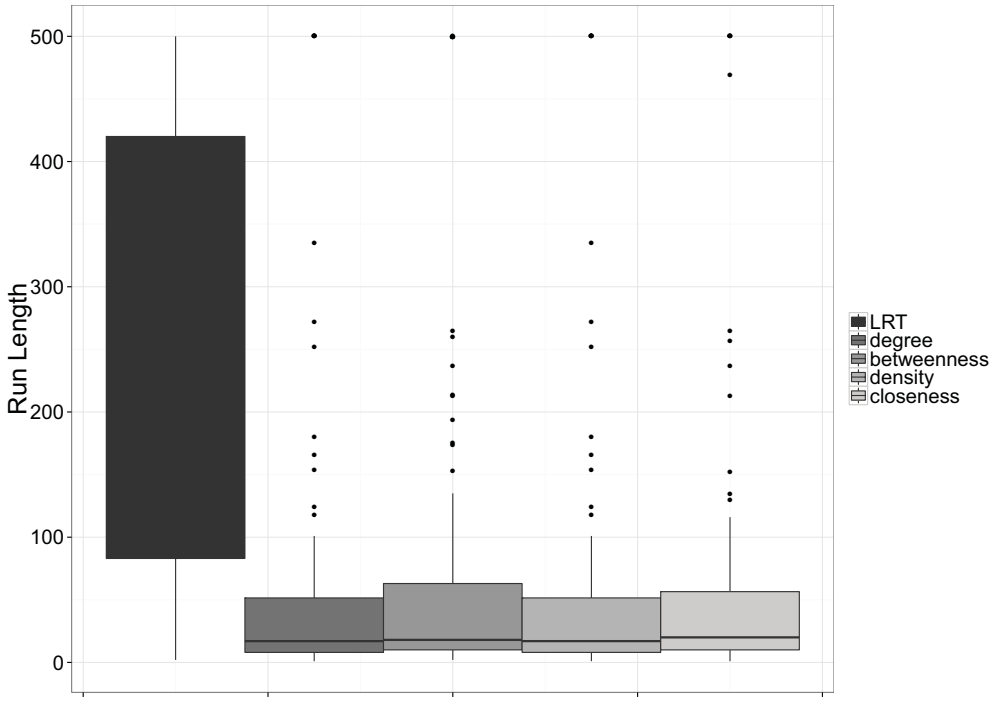


Figure 4.10: Run Lengths of the Different Procedures Under No Change.

Change	δ_1	δ_2
c_1	0.3	-0.3
c_2	0.3	-0.5
c_3	0.3	0.3
c_4	0.3	0.5
c_5	0.3	0

Table 4.1: The Induced Changes Of the Experiment.

RL of 50 is recorded. Figure 4.11 summarize the RLs for networks with $\nu = 50$ and $\nu = 100$ in Part (a) and Part (b) respectively. In these figures, each box plot depicts the RLs of the 100 streams for each case with the horizontal axis showing the change as in Table 4.1. These results provide evidence for the strength of the proposed approach.

To investigate the differences between the procedures, consider first the two changes

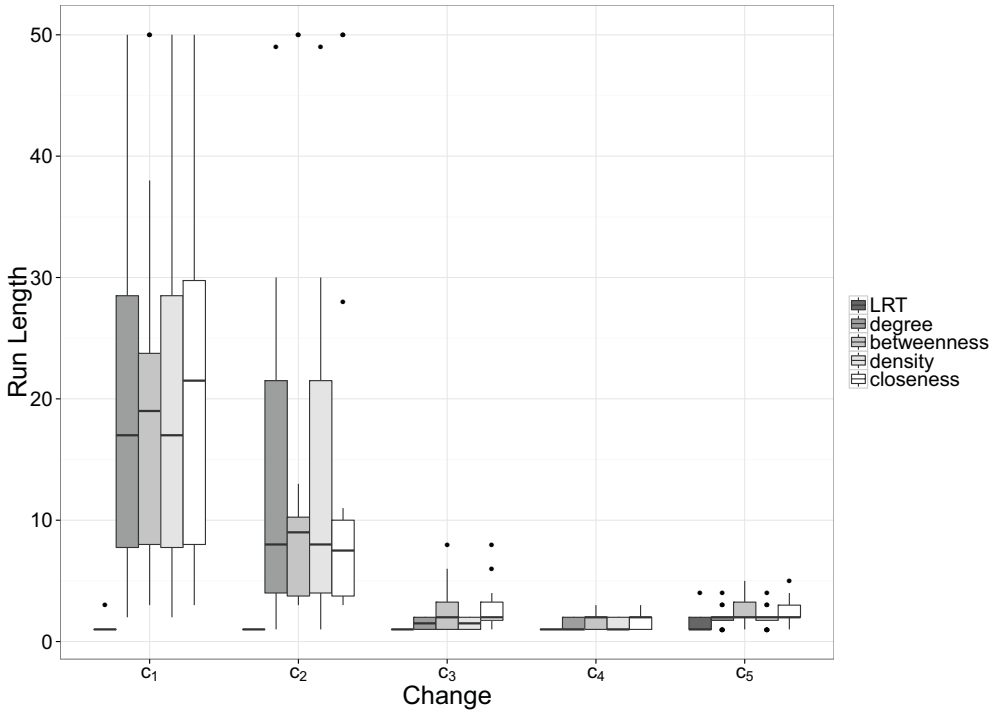
c_3 and c_4 where the LRT approach performs similar to the CUSUMs. Both these changes induce an increase in edge probability for all vertex pairs and, therefore, lead to higher connectivity over the entire network (similar to the change in Part (b) of Figure 4.1). This change is reflected on the network measures (such as average degree) enabling detection through CUSUM charts on the measures. The proposed method is also able to detect such a change due to the presence of edges that have been assigned a low probability under the null hypothesis of no change.

More subtle changes are induced through c_1 and c_2. These changes induce a local increase in edge probability between vertex pairs with $x_{2ij} = 1$ and reduced edge probability otherwise. Such a change is not reflected well on network measures based solely on the network topology, hindering the change detection. The proposed method, however, is able to detect the temporal inhomogeneity in the $X_2 = 1$ region of the attribute space.

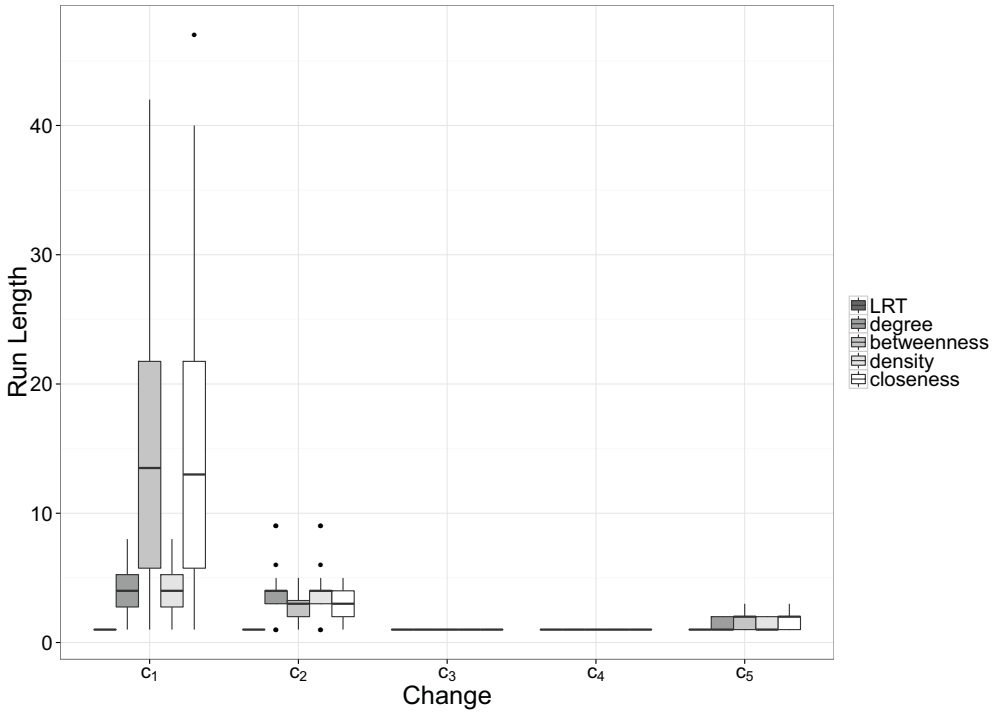
4.6 Conclusion

The dynamics of relationships between entities in complex, real world systems generate network streams. This chapter proposes an extension of statistical monitoring to such streams. Unlike current methods that are based on measures from the network topology, the proposed method monitors the underlying network formation mechanism via vertex attributes. This provides a flexible method, able to detect different forms of anomaly that arises from different network edge formation mechanism.

The next chapter continues network monitoring towards the development of diagnostic tools to shed light on the anomaly upon detection (Deng *et al.*, 2012; Li *et al.*, 2008; Runger *et al.*, 1996) as well as methods that can handle various network attributes (such as vertex and edge attributes).



(a)



(b)

Figure 4.11: Run Lengths of the Different Procedures for Different Changes.

Chapter 5

MONITORING TEMPORAL HOMOGENEITY IN NETWORK STREAMS WITH SUPERVISED LEARNING

5.1 Introduction

Networks provide a rich model for entity interactions in many complex systems such as social, cyber and biological systems. The intrinsic dynamics of interactions in such systems is an important issue that needs to be captured and monitored. The dynamics of communication logs over time is, for example, better captured through a stream of network snapshots compared to a single static network. Monitoring such streams is an important problem as changes in the system are likely reflected in the network. This has motivated research on network monitoring towards detecting networks with anomalies with respect to past networks. Network monitoring for anomaly detection is usually tailored around two objectives that we define as testing for *static homogeneity* and testing for *temporal homogeneity*. Considering a network as a set of *transactions* that describe the interactions between system entities, testing for static homogeneity aims to detect networks that have anomalous transactions within the current network (see for example Miller *et al.* (2013)). Testing for temporal homogeneity, on the other hand, aims to detect networks that have anomalous transactions with respect to past networks. This is an important problem as changes in the system are likely reflected in the transactions that compose the network and is the focus of this chapter.

To illustrate, consider a hypothetical example of an institution's email network. The vertices represent the employees and edges represent emails between employees.

The vertices and connecting edges induce the topological structure of the network as depicted in the left of Figure 5.1. Most current network monitoring approaches for temporal homogeneity are restricted to only the topological structure of the network. A typical approach is to monitor extracted measures from the topological structure through time. As an example, McCulloh and Carley (2011) constructed control charts over different network measures such as density, average degree, average closeness and average betweenness. The work by Priebe *et al.* (2005), Marchette (2012) and Neil *et al.* (2014) monitored scan statistics for this purpose. This involves enumerating fixed, defined, windows over the entire network to extract measures of the structure. Similarly, the work by Park *et al.* (2013) used a fusion of network measures (including the scan static).

Many real networks are associated with additional layers of data provided through associated vertex and edge attributes. In the hypothetical example above, possible vertex attributes are the role and work experience of employees and possible edge attributes are the size and topic of emails. An interaction between two employees can then be described through these attributes (e.g. the role of the sender, the topic, etc). The vector of these attributes is defined as a transaction. Note that the transaction may also include attributes from the network structure (topological attributes) such as the number of other emails the sender sends to other employees (the origin vertex degree). The collection of such transactions gives rise to a *multi-dimensional* network such as the network in right of Figure 5.1. This is the same network as Part (a) but now augmented with additional attributes. Specifically, the role of the employee is depicted through color, the associated department through shape and experience level through the size. Also, the topic of the email is depicted through the color of the edge and its size through the width of the edge.

This chapter proposes a method to monitor a stream of multi-dimensional net-

works. Specifically, we consider a stream of network snapshots where at each time stamp a separate network that is composed of a connected set of transactions is obtained from the system. Our focus is monitoring the different attributes of transactions (monitoring the color and width of the edges between pairs of vertices with different color, shape and size combinations in the example). The detection of an increase in emails from employees of a department serves as a simple example.

The work by Priebe *et al.* (2010) considered monitoring networks where each edge is associated with an attribute. This is, however, limited to a single categorical attribute. Also, the work presented in Chapter 5 considered (only) vertex attributes in network monitoring. Many real networks, however, are augmented with both vertex and edge attributes. Monitoring such networks requires monitoring the joint distribution of the attributes of transactions which has received little attention in the literature and is our focus.

Monitoring a stream of such *multi-dimensional* networks calls for a method to detect change in any region defined by the attributes of the transactions. An important issue here is the high dimensionality that arises from transactions having a large number of attributes. Simultaneous monitoring of the regions is defeated by the combinatorial explosion of the number of region subsets making this problem especially challenging. Nonetheless, there are myriads of applications where monitoring different dimensions of transactions is needed. In addition to monitoring email networks, logistic networks are another example where monitoring the size and type of packages between cities with different population and climate is of interest. Other examples include biological networks where monitoring connections between genes and proteins with different properties is of interest. It should be noted that *high-dimensional monitoring* has important application in other non-network related problems (Dávila *et al.*, 2011).

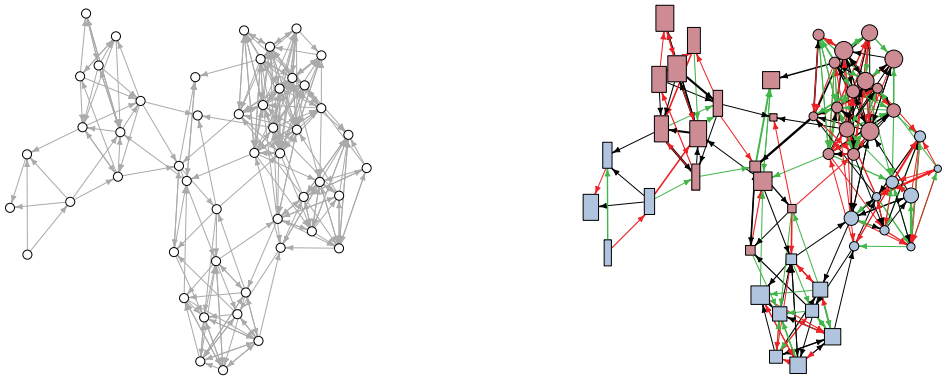


Figure 5.1: The Topological Structure of an Example Network is Depicted on the Left. The Same Network is Augmented with Vertex and Edge Attributes in the Right. Each Vertex is Associated with Three Attributes Depicted Through Color, Shape and Size (Vertex Attributes) and Each Edge with Two Attributes Depicted Through Color and Width (Edge Attributes). Additional Attributes, Such as the Degree of the Origin Vertex, May Be Defined from the Network Topology (Topological Attributes). Each Transaction is Then Defined as the Vector of Vertex, Edge and Topological Attributes.

A further complexity in network monitoring is the scope of change (i.e. global vs local change). A global change affects all the transactions on the network, while a local change affects only a small subset of the transactions (referred to as *partial temporal inhomogeneity*). Different applications present changes of different scope. For example, an event (e.g. policy change) that affects the communication of all employees will likely impose a change on all transactions on the network, while an event that affects only employees of a certain role will likely impose a change that only affects a subset of the transaction. A monitoring approach should be sensitive to both these types of changes.

By presenting each transaction as a vector of its attributes, transactions from the current network are contrasted to a set of reference transactions that characterize typical network behavior through a supervised leaner. The idea is the transform of network monitoring to supervised learning that provides a set of powerful tools that

are used towards devising a monitoring approach that effectively detects change in any region defined by the transactions' high dimensional feature space that affects only a small subset of the transactions. Moreover, diagnostic tools that provide insight on the nature of change are derived. The details of the proposed method are described in Section 5.2. This is followed by Sections 5.3 that describes our experimental evaluation based on synthetic and real networks. Finally, Section 5.4 provides concluding remarks.

5.2 Multi-Dimensional Network Monitoring

We start this section with our data presentation for multi-dimensional networks. A network is composed of a set of vertices and edges. The vertices represent system entities (e.g., employees of an institution) and edges represent interactions between system entities (e.g. emails between pairs of entities). Interactions between entities can be characterized by a set of attributes. We define a *transaction*, denoted by e_i, between vertices i' and i'', as an M dimensional vector of attributes that describes the interaction. We review different types of attributes next.

Vertex attributes: These are properties of the system entities (modeled as vertices) that the interaction (modeled as an edge) flows between. In the earlier hypothetical example, the roles of the sender and receiver of the email are examples of vertex attributes.

Edge attributes: These are properties of the interactions between system entities. The size and topic of an email serve as examples.

Topological attributes: These are properties of the interaction with respect to other interactions on the network. The sender's vertex degree (number of other emails the sender sent) is an example of such attributes.

At each discrete time stamp t, a network snapshot $E(t)$, that is composed of a set

of transactions e_i is obtained from the system. That is,

$$E(t) = \{e_i(t); i = 1, \cdots, N_t\} \tag{5.1}$$

where N_t denotes the total number of transactions at time t. Now, given a stream of multi-dimensional network snapshots $E(t), t = 1, 2, \cdots$, our objective is to detect temporal inhomogeneity in $E(t)$. This is a challenging problem as simultaneous monitoring of individual attributes is defeated by the combinatorial explosion of the number of region subsets. We next provide the details of the proposed method for this problem.

5.2.1 Network Monitoring as a Supervised Learning Problem

Our method for monitoring multi-dimensional networks is based on the idea of transforming a monitoring problem to one of supervised learning. This transformation provides a powerful set of tools that may be used to address important problems in network monitoring. We start by assuming a reference transaction set that characterizes typical network behavior, denoted by $E(0)$. The transactions in the reference set are considered to be a sample from an unknown distribution $f_0(e)$. At time $t = \tau$, a change is considered to be present if transactions in $E(\tau)$ follow a distribution other than $f_0(e)$, denoted by $f_1(e)$, which we are interested in detecting. The generalized likelihood ratio (GLR) principle may be used as a guide for change detection (Fan *et al.*, 2001). In this direction, the problem is formulated as testing the following hypothesis

$$H_0 : e_i \sim f_0(e_i), \forall e_i \in E(0) \cup E(\tau)$$
$$H_1 : e_i \sim f_0(e_i), \forall e_i \in E(0);\ e_i \sim f_1(e), \forall e_i \in E(\tau)$$

which results in the GLR test statistic

$$\Lambda(\tau) = \dot{\sum_{e_i \in E(\tau)}} \log \frac{f_1(e_i)}{f_0(e_i)} \tag{5.2}$$

This test statistic assumes knowledge of the distributional forms of $f_0(e)$ and $f_1(e)$ which is often an unrealistic assumption. Relaxing this requirement, our method tests for change through the transform to a supervised learning problem. The idea is to contrast transactions in $E(\tau)$ to transactions in $E(0)$ through a supervised learner. Towards this end, each transaction is labeled with a class attribute y according to the following recipe

$$y_i(t) = \begin{cases} 0 & \text{if } e_i(t) \in E(0) \\ 1 & \text{if } e_i(t) \in E(\tau) \end{cases} \tag{5.3}$$

A supervised learner is then constructed from the two classes and contrasts the transactions in the current time stamp τ to those in the reference set. In case of temporal inhomogeneity, the transactions in $E(\tau)$ and $E(0)$ follow different distributions which heightens the discrimination strength of the learner. This idea can be used towards devising monitoring statistics that measure the learner's discrimination strength based on the notion that if the learner can classify correctly, the network has indeed changed. Therefore, high values of the monitoring statistics should indicate high discrimination strength of the learner between transactions in $E(0)$ and transactions in $E(\tau)$ and, thereby, indicate the presence of change.

5.2.2 Monitoring Statistics

We next devise a set of monitoring statistics that are used towards monitoring decisions. Different monitoring statistics can be considered to measure the learner's discrimination strength between transactions in $E(\tau)$ and $E(0)$. The learner's error rates are indicative of the discrimination strength and can, thus, be used towards

monitoring decisions. At a finer grain, one can use the class probability estimates, if provided by the learner, to gain insight on its discrimination strength. We use $\hat{p}_c(e_i)$ to denote the class probability estimate for e_i belonging to class $c \in \{0, 1\}$ and omit the time index t in the notation for simplicity in the rest of this chapter. The mean $\hat{p}_0(e_i)$ in $E(0)$ and the mean $\hat{p}_1(e_i)$ in $E(\tau)$ are considered as two monitoring statistics shown below

$$AP_0(\tau) = \frac{\dot{\sum}_{e_i \in E(0)} \hat{p}_0(e_i)}{N_0} \tag{5.4}$$

$$AP_1(\tau) = \frac{\dot{\sum}_{e_i \in E(\tau)} \hat{p}_1(e_i)}{N_\tau} \tag{5.5}$$

Monitoring statistics AP_0 and AP_1 are possible choices but we focus on the GLR test statistic as a guide to derive other monitoring statistics. Denoting the prior probability of a transaction belonging to class 1 by π, Bayes rule yields the following

$$p(e_i \mid c = 0) = \frac{p(e_i)p_0(e_i)}{1 - \pi} \tag{5.6}$$

$$p(e_i \mid c = 1) = \frac{p(e_i)p_1(e_i)}{\pi} \tag{5.7}$$

The proportion of likelihoods in the GLR statistic Λ may be replaced by the proportion of probabilities and Equations 5.6, 5.7 and the learner's class probability estimates may be used to estimate Λ. In this direction, reference Deng *et al.* (2012) uses the following statistic.

$$LR(\tau) = \dot{\sum_{e_i \in E(\tau)}} \log \frac{\hat{p}_1(e_i)}{\hat{p}_0(e_i)} \tag{5.8}$$

We note that in addition to replacing the proportion of the likelihoods with proportion of probabilities, the derivation of LR involves taking the logarithm. This is common

practice since it is usually more convenient to work with the logarithm of the likelihood function.

An important consideration in network monitoring is the scope of change (i.e. global vs local change). A global change affects all transactions on the network, while a local change affects only a small subset of the transactions. We refer to local changes as *partial temporal inhomogeneity* in this chapter. Different applications present changes of different scope and a monitoring statistic should be sensitive to both these types of changes. This motivates the use of a mixture distribution for $f_1(e)$. Under this model, we have

$$f_0(e) \sim g_0(e)$$
$$f_1(e) \sim (1-\pi)g_0(e) + \pi g_1(e)$$

that results in the following GLR statistic

$$\Psi(\tau) = \sum_{e_i \in E(\tau)}^{\cdot} \log \frac{(1-\pi)g_0(e_i) + \pi g_1(e_i)}{g_0(e_i)} \tag{5.9}$$

The distributional form of $g_0(e)$ and $g_1(e)$, as well as estimates for their parameters are needed for calculating this statistic. By assuming knowledge of the distributional form, the Expectation Maximization (EM) algorithm (Dempster *et al.*, 1977) may be adopted for estimating the parameters of $g_0(e)$ and $g_1(e)$ and, thereby, estimating Ψ. The EM algorithm considers unobserved latent variables and estimates the parameters of $g_0(e)$ and $g_1(e)$ through an iterative approach. We propose an iterative method based on supervised learning that also considers latent variables for estimating Ψ. This method does not, however, assume knowledge of the distributional form of $g_0(e)$ and $g_1(e)$ and yields an estimate for Ψ without estimating parameters of the distribution.

The proposed method is based on unobserved latent variables z_i that take a value

of 0 if e_i follows $g_0(e)$ and a value of 1 otherwise. Assuming knowledge of z_i, Equation 5.9 can be written as

$$\dot{\sum_{e_i \in E(\tau)}} [z_i \log g_1(e_i) - z_i \log g_0(e_i) + z_i \log \pi + (1 - z_i) \log(1 - \pi)] \tag{5.10}$$

Replacing proportion of likelihoods with proportion of probabilities and using Equations 5.6, 5.7 and the class probability estimates, Equation 5.10 is written as

$$LRP(\tau) = \dot{\sum_{e_i \in E(\tau)}} z_i \left[\log \hat{p}_1(e_i) - \log \hat{p}_0(e_i)\right] + N_\tau \log(1 - \pi) \tag{5.11}$$

The z_is are actually unknown and are treated as missing values. For calculating LRP, the proposed iterative method alternates between performing an expectation step that assigns a class y_i to each e_i, and a training step that constructs a classier using the current class assignments from the expectation step. The classifier is used to assign y_i in the next expectation step. This procedure is iterated until convergence and the classes y_i at the last iteration are used to replace z_i in LRP. The details are provided next.

In the initial iteration all transactions in $E(\tau)$ are labeled as class 1 and all transactions in $E(0)$ as class 0. Using superscripts to refer to iteration, we have

$$y_i^{(0)} = \begin{cases} 0 & \text{if } e_i \in E(0) \\ 1 & \text{if } e_i \in E(\tau) \end{cases} \tag{5.12}$$

A supervised learner is then trained on $E(0) \cup E(\tau)$ with class y values according to Equation 5.12. Each consequent iteration $k, k = 1, \cdots, K$ starts by assigning class labels to transactions in $E(\tau)$ according to

$$y_i^{(k)} = \begin{cases} 0 & \text{if } \hat{p}_1^{(k-1)}(e_i) < \rho \\ 1 & \text{otherwise} \end{cases} \tag{5.13}$$

This step is similar to EM's Expectation step: transactions are re-classified using the current learner. In each expectation step, a new class is assigned to each e_i. At iteration k, consider the set of e_i that have been classified as class 1 by all previous iterations. Using $E_1^{(k)}(\tau)$ to denote this set, we can write

$$E_1^{(k)}(\tau) = \left\{ e_i \mid e_i \in E(\tau); y_i^{(j)} = 1, \forall j < k \right\} \tag{5.14}$$

Now, a supervised learner is constructed on $E_1^{(k)}(\tau) \cup E(0)$ and the next iteration follows. Training the learner based on the current class assignment is somewhat similar to EM's Maximization step: the learner is trained using the current classes and, therefore, its classification conforms to the current class assignments just as the maximization step of EM estimates parameters that conform to the current latent variables. In the case of training with a tree classifier, for example, this translates to the tree partitioning the feature space in a way that best (greedily) conforms to the current classes.

Using maximum likelihood estimation (MLE) (Casella and Berger, 1990), each expectation step yields an estimate for π

$$\hat{\pi}^{(k)} = \frac{\left| \hat{E}_1^{(k)}(\tau) \right|}{N_\tau} \tag{5.15}$$

The iterations are repeated until no transaction from $E(\tau)$ is consistently classified as class 1 (i.e. $\left| E_1^{(k)}(\tau) \right| = 0$) or in case of classification consistency of two consequent iterations on the current transactions (i.e. $E_1^{(k)}(\tau) = E_1^{(k-1)}(\tau)$). We use K to denote the index of the last iteration. The number of iterations to reach convergence is problem-specific but is always reached since one of the two mentioned criteria will be met after some iterations.

By construction, monitoring statistics AP_0, AP_1, LR are useful for detecting temporal inhomogeneity that exhibits on the entire current network. Monitoring statistic

LRP take the partial temporal inhomogeneity into account and is, thus, better suited for situations where only a subset of the network transactions are under change.

As a final note in this section, it should be mentioned that the GLR principle has been used as a guide for devising monitoring statistics because it is a general and powerful method for hypothesis testing in many problems (Fan *et al.*, 2001). Nevertheless, other measures for difference in probability distributions, such as the Kullback-Leibler divergence (Pardo, 2005), are also possible.

5.2.3 Supervised Learner

The proposed method uses a supervised learner to contrast the transactions in $E(\tau)$ to transactions in $E(0)$. In general, any learner that can handle the complexities of multi-dimensional network monitoring may be used. These complexities are briefly discussed next.

High-dimensional transactions with disparate attributes: Each transaction in a multi-dimensional network may be associated with a large number of attributes of disparate type (numerical and categorical) and disparate scales. Vertex attributes, for example, can include both numerical (such as the user's age) and categorical (such as the user's gender) attributes.

Unbalanced edge sets: The reference set is collected over consecutive time periods to characterize typical behavior and is, thus, expected to be larger in size compared to the current network (i.e. $N_0 > N_\tau$). The adopted learner should, therefore, handle the unbalanced class problem.

Class probability estimates: The learner should provide class probability estimates that can be used towards monitoring decisions.

Nonlinearities: Monitoring multi-dimensional networks involves monitoring the joint distribution of the transaction's attributes and should handle nonlinearities be-

tween the attributes.

The Random Forest classifier (RF) (Breiman, 2001) is adopted as the supervised learner since it can accommodate the mentioned complexities. RF constructs a collection of trees on bootstrapped data so that a diverse ensemble of trees is produced.

In our application, the RF at each iteration is constructed on $E_1^{(k)}(\tau) \cup E(0)$ using the assigned y values. The constructed data $(E_1^{(k)}(\tau) \cup E(0))$ that the RF is constructed on is likely to be imbalanced since $N_0 > N_\tau$ in most situations. Similar to Deng *et al.* (2012), a stratified sampling approach is used to handle the unbalanced class problem that arises due to the difference of N_τ and N_0. Let $n_1^{(k)}$ and $n_0^{(k)}$ denote the number of transactions with $y_i^{(k)} = 1$ and $y_i^{(k)} = 0$ that is used to construct each tree at iteration k. In the initial iteration, stratified sampling is used to set $n_1^{(0)} = n_0^{(0)} = N_\tau$. The RF is trained and class probability estimates are obtained. Monitoring statistics AP_0, AP_1 and LR are constructed based on $\hat{p}_c^{(0)}$, $c \in \{0, 1\}$. For the LRP statistics, we proceed with the iterative method and use stratified sampling with $n_1^{(k)} = \left|E_1^{(k)}(\tau)\right|$ and $n_0^{(k)} = N_\tau - \left|E_1^{(k)}(\tau)\right|$. Class probability estimates $\hat{p}_c^{(K)}$, $c \in \{0, 1\}$ and $\hat{\pi}^{(K)}$ are used for calculating LRP. The details of using RF to calculate monitoring statistics AP_0, AP_1, LR and LRP is summarized in Algorithm 1 called Iterative Forest Monitoring. We note that $\rho = 0.5$ is used in our implementation. Further study can be done to evaluate the choice for this parameter.

5.2.4 Temporal Inhomogeneity Diagnostics

Upon detecting temporal inhomogeneity, insight on the nature of change is important. This is similar to the fault diagnosis problem encountered in multivariate process monitoring (Runger *et al.*, 1996). The proposed method leverages the variable importance (VI) measures provided by the RF (Breiman *et al.*, 1984). An increase in these measures provide information about the temporal inhomogeneity. Increase in

Algorithm 1: Iterative Forest Monitoring

Initialization

0.1. Class assignment

$$y_i^{(0)} = \begin{cases} 0 & \text{if } e_i \in E(0) \\ 1 & \text{if } e_i \in E(\tau) \end{cases}$$

0.2. Train RF on $E(\tau) \bigcup E(0)$. Use stratified sampling $n_1^{(0)} = n_0^{(0)} = |E^\tau|$.

While $\left|E_1^{(k)}(\tau)\right| \neq 0$ and $E_1^{(k)}(\tau) \neq E_1^{(k-1)}(\tau)$

Iteration $k = 1, \cdots$

k.1. Class assignment

$$y_i^{(k)} = \begin{cases} 0 & \text{if } \hat{p}_1^{(k-1)}(y_i) < \rho \\ 1 & \text{otherwise} \end{cases}$$

k.2. Train RF on $E_1^{(k)}(\tau) \cup E(0)$. Use stratified sampling $n_1^{(k)} = \left|E_1^{(k)}(\tau)\right|$ and $n_0^{(k)} = N_\tau - \left|E_1^{(k)}(\tau)\right|$.

---

different attributes is indicative of different information about the change (elaborated in the case studies of Section 5.3).

The iterative forest monitoring algorithm allows for the calculation of the variable importance measures at different iterations. We use $VI^{(k)}(a)$ to denote the VI of attribute a at iteration k. In case of partial temporal inhomogeneity, the VI of initial iterations will likely not be useful for diagnostics. The explanation follows: the goal of the iterations is to *sieve* through the transaction in $E(\tau)$ so that we are left with a better estimate of the set of transactions in $E(\tau)$ that is impacted by the change. Using the formulation presented in Section 5.2.2, this is set $\{e_i \mid e_i \in E(\tau); z_i = 1\}$

which we estimate using $E_1^{(K)}(\tau)$. At the last iteration K, we contrast transactions in $E_1^{(K)}(\tau)$ and $E(0)$ so that the RF splits on attributes that actually discriminate the subset of transactions that are affected by the change to transactions in the reference set. Therefore, the VI at last iteration K is likely more accurately indicate the important attributes that contribute to local change.

5.3 Experimental Evaluation

The proposed method is illustrated through case studies in this section. Both synthetic and real networks are considered. Synthetic networks where the ground truth concerning the change is known are used to allow for the evaluation of the method.

In section 5.3.1, monitoring networks with both vertex and edge attributes are considered. We note that no current method considers monitoring such networks. In section 5.3.2, monitoring networks without vertex and edge attributes, where only the network topology is available, is compared to an alternative method. Section 5.3.3 considers monitoring the Enron network using both vertex and edge attributes. This is the first work to study monitoring the Enron network by integrating vertex and edge attributes. Finally Section 5.3.4 considers the sensitivity of the LRP statistic for detecting partial temporal inhomogeneity.

5.3.1 Networks with Vertex and Edge Attributes

Network generation scheme: We first provide a description of how each network in the stream is generated. The generation scheme is inspired by the Erdos-Renyi random graph model that describes networks where edges are formed independently between each pair of vertices with a common probability (Erdos and Renyi, 1959). This is an overly simple model and various attempts have been made to model sys-

tematic deviations from pure randomness. As an example, the stochastic blockmodels (SBM) (Wang and Wong, 1987) is a multi-class extension of the Erdos-Renyi model where stochastic equivalence is assumed among vertices in the same class and edges are formed conditionally independent given the class membership of the vertex. Another model, is the attributed network model that assumes the probability of an edge between two vertices is a function of a set attributes (Miller *et al.*, 2013). Note that this is an extension of the Erdos-Renyi and SBM and provides a flexible network model based on network attributes that have been shown to be useful for various network modeling tasks such as link prediction and attribute inference (Al Hasan *et al.*, 2006; Gong *et al.*, 2011; Kim and Leskovec, 2010; Kumar *et al.*, 2004). The details of the generation scheme is provided next.

Networks with 50 vertices are considered. Each vertex is associated with three vertex attributes. Two are Uniform and one is Bernoulli. That is, $\alpha_1 \sim Uniform(12, 36)$, $\alpha_2 \sim Bernoulli(0.5)$ and $\alpha_3 \sim DiscreteUniform(1, 3)$. These attributes are depicted through the size, color and shape of the vertices, respectively, in the subsequent figures. Let $\alpha_{ji'}$ and $\alpha_{ji''}$ denote the value of attribute α_j for the origin (vertex i') and destination (vertex i'') of edge e_i respectively, then the following attributes are defined from vertex attributes for e_i.

$$a_{1i} = \text{ABS}(\alpha_{1i'} - \alpha_{1i''}) \tag{5.16}$$

$$a_{2i} = \text{NXOR}(\alpha_{2i'}, \alpha_{2i''}) \tag{5.17}$$

$$a_{3i} = \text{ABS}(\alpha_{3i'} - \alpha_{3i''}) \tag{5.18}$$

The ABS function returns the absolute value and the NXOR function returns the logical complement of the exclusive disjunction. Additionally, each transaction is associated with a size described through attribute $a_4 \sim Normal(1, 1)$. This attribute is depicted through the edge width graphically. A transaction is defined as the four

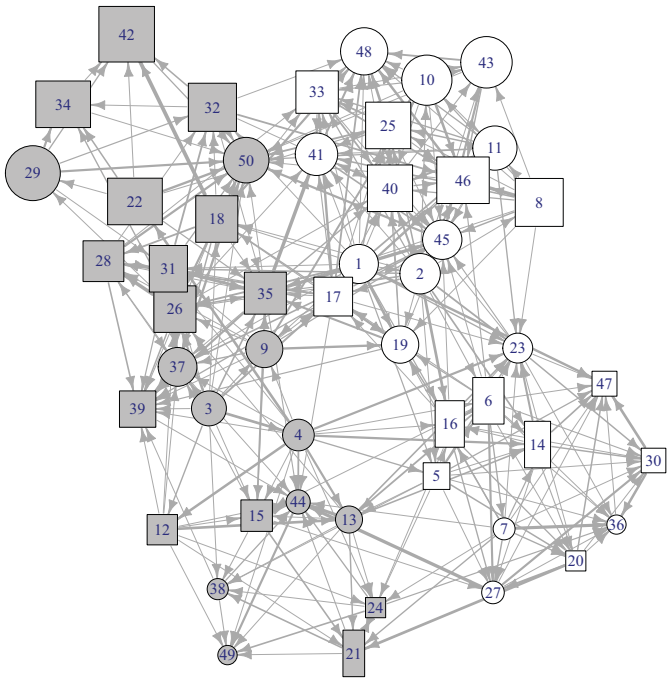


Figure 5.2: An Example Network Under Typical Conditions. Each Vertex is Associate With Three Vertex Attributes that are Depicted Through the Size, Color And Shape Of The Vertex And Each Edge Is Associated With a Transaction Size That Is Depicted Through Its Width. Typically Edges Are Formed Between Vertices Of Similar Color And Size And Transaction Size Follows The same Normal Distribution on The Entire Network.

dimensional vector (a_1, a_2, a_3, a_4). The number of transactions e_i between two vertices i' and i'' is then modeled as $Binomial(7, p_i = \text{logit}^{-1}(-2 - 0.5a_{1i} + 3a_{2i}))$. This promotes edges between vertices of similar size and color. We consider monitoring transactions of different width between vertices with different size, color and shape combinations. Figure 5.2 demonstrates a network under typical conditions that was generated according to the described scheme. The reference set $E(0)$ is constructed from transactions generated according to this scheme.

We first consider a change that promotes transactions of larger size between ver-

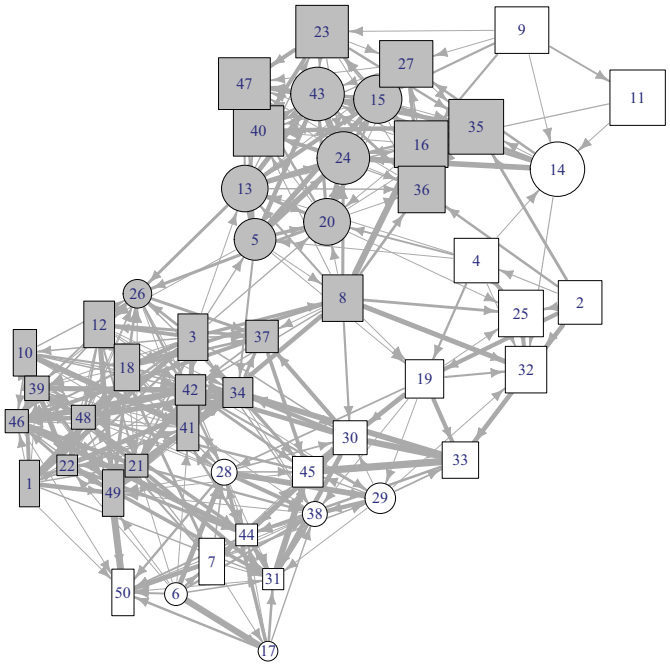


Figure 5.3: An Example of Temporal Inhomogeneity on the Network. Larger Sized Transactions (Larger Width Edges) are Observed Between same Shaped Vertices.

tices of the same shape. The change concerns a shift in the mean of this distribution such that its distribution changes from $Normal(1, 1)$ to $Normal(4, 1)$ between vertices of the same shape. Figure 5.3 depicts a network under such change. Note that this change affects many of the transactions and can, therefore, be considered as a global change.

To demonstrate the detection of such change, we monitor the statistics AP_0, AP_1, LR and LRP after inducing a change at $t = 15$. Figure 5.4 depicts the results. The change is clearly detected through all four monitoring statistics.

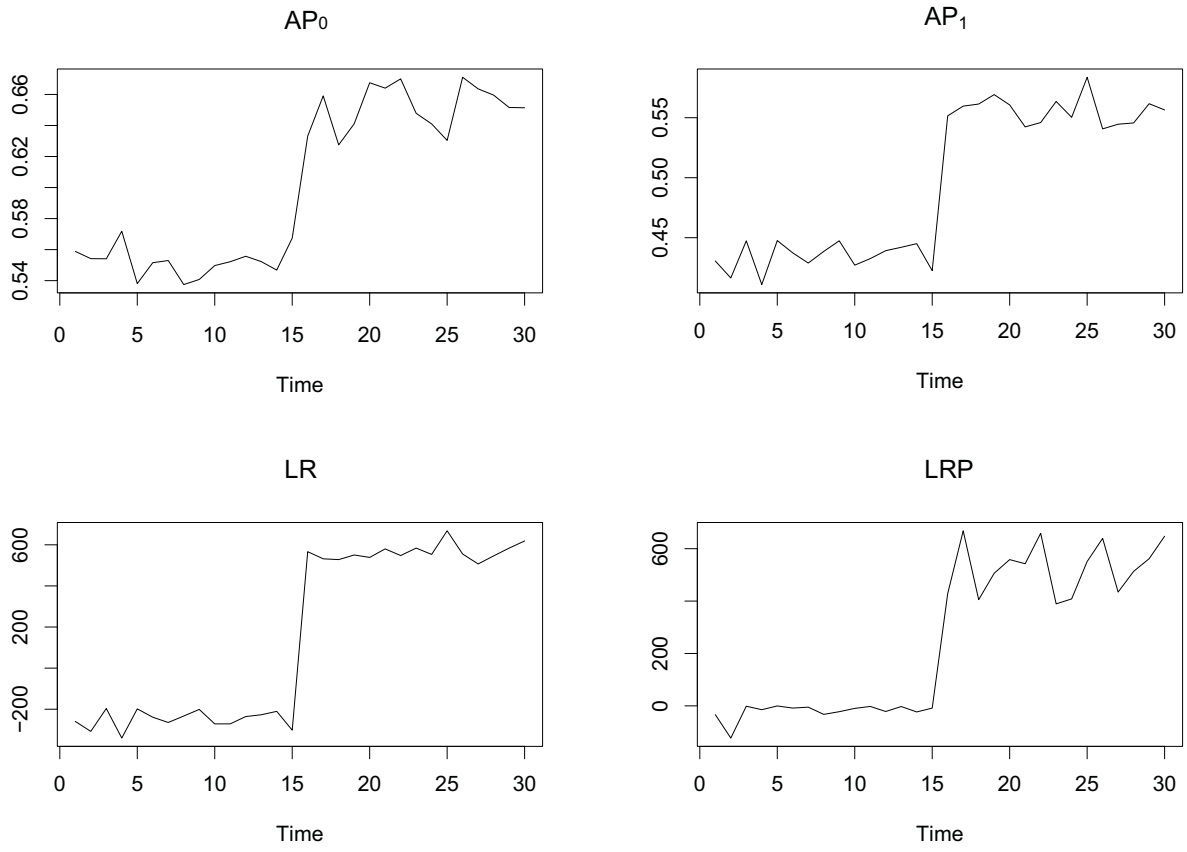


Figure 5.4: Plot of Different Monitoring Statistics Versus Time to Detect a Change in the Transaction Size Between Same Shaped Vertices. The Change is Clearly Detected by All Four Monitoring Statistics.

Next, the VI measures are used to provided change diagnostics. Figures 5.5 and 5.6 plot the VI of the iterative forest monitoring algorithm at the initial (iteration 0) and last (iteration K) iteration versus time, respectively. Increase in VI measures for attribute a_3 (vertex shape) and a_4 (edge width) correctly identify the nature of change. We note that in this case study, the VI at the initial and last iteration convey similar diagnostics. This is due to the global nature of the change. A decrease in $VI^{(0)}(a_1)$ and $VI^{(0)}(a_2)$ is also evident. An explanation follows: before the change, the transactions with different class labels actually follow the same distribution. Therefore, the class assignment does not discriminate the transactions' distribution and

can, hence, be regarded as arbitrary. Still, the supervised problem is presented and RF splits on attributes which results in a moderately high VI for a_1 and a_4. After the change, the RF splits on attributes that actually discriminate transactions from different distributions which results in an increase of the importance of a_3 and a_4.

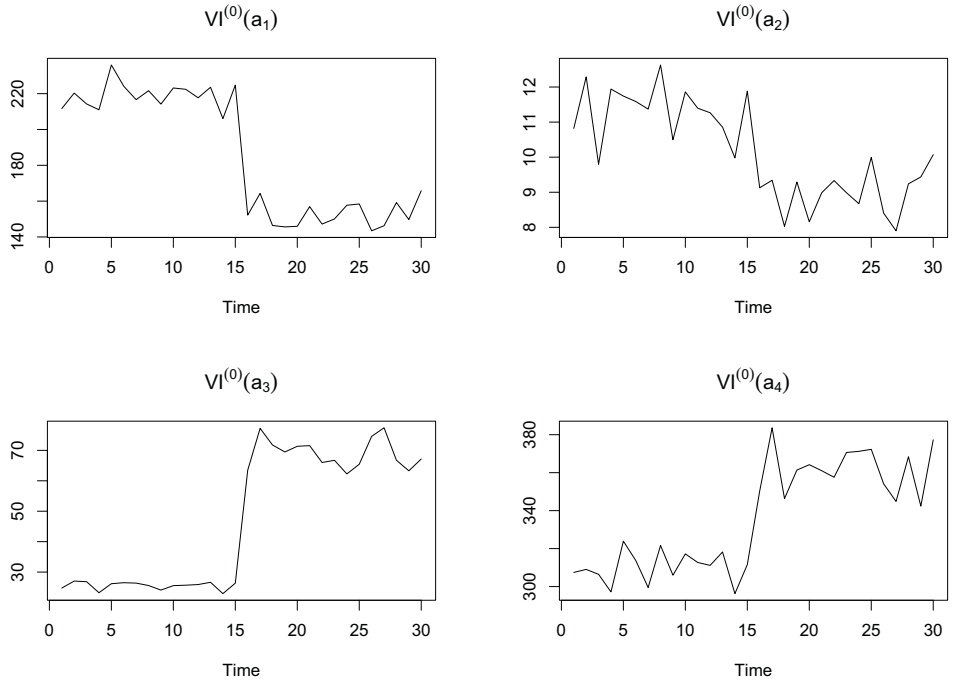


Figure 5.5: Plot Of Variable Importance at Iteration 0 of the Iterative Forest Algorithm ($VI^{(0)}$) Versus Time. A Change is Induced at $t = 15$ that Distorts the Transaction Size (Attribute a_4) Between Vertices of Same Shape (Attribute a_3) and is Detected Through the Monitoring Statistics. The Nature of the Change is Identified Through Increase in VI Measures for Attribute a_3 (Vertex Shape) and a_4 (Edge Width).

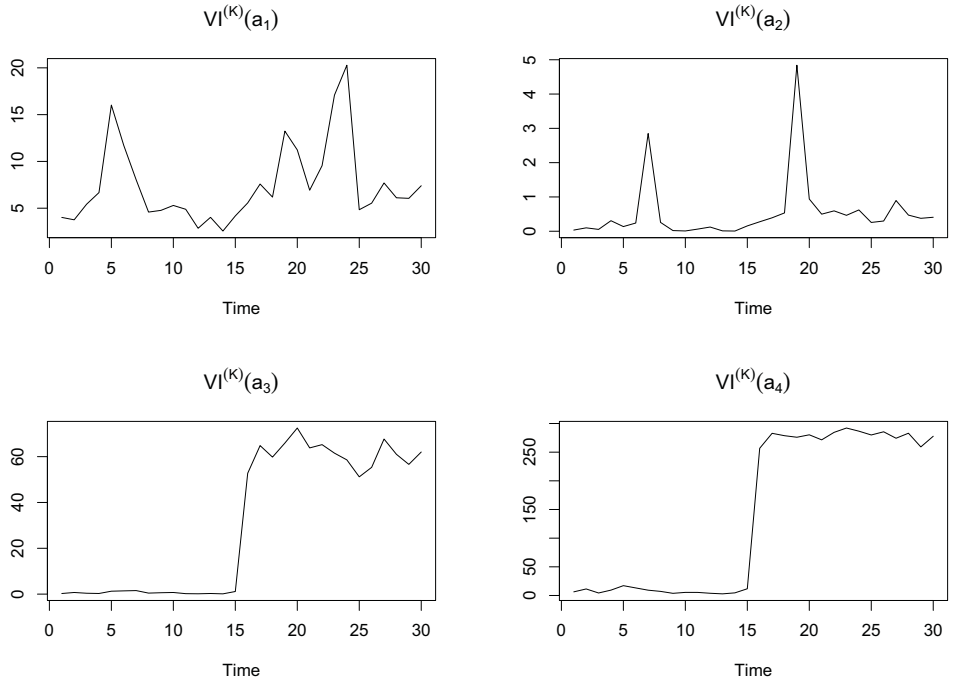


Figure 5.6: Plot of Variable Importance at Iteration K (Last Iteration) of the Iterative Forest Algorithm ($VI^{(K)}$) Versus Time. A Change is Induced at $t = 15$ that Distorts the Transaction Size (Attribute a_4) between Vertices of Same Shape (Attribute a_3) and is Detected Through the Monitoring Statistics. The Nature of the Change is Identified Through Increase in VI Measures for Attribute a_3 (Vertex Shape) and a_4 (Edge Width).

We next consider a change that promotes transactions of larger size over a small subset of the vertices (five of the vertices). Specifically, the transaction size follows a $Normal(1, 1)$ for all transaction except for transactions between the five vertices that follows $Normal(4, 1)$ distribution. This is a local change as it only affect a subset of the transactions (transactions between five vertices). Figure 5.7 depicts a network under this change.

To demonstrate the detection of this change, we consider monitoring the introduced statistics after inducing a change at $t = 15$ (see Figure 5.8). We note that this change is an example of partial temporal inhomogeneity and is, thus, better detected by monitoring statistic LRP that takes partial temporal inhomogeneity into account.

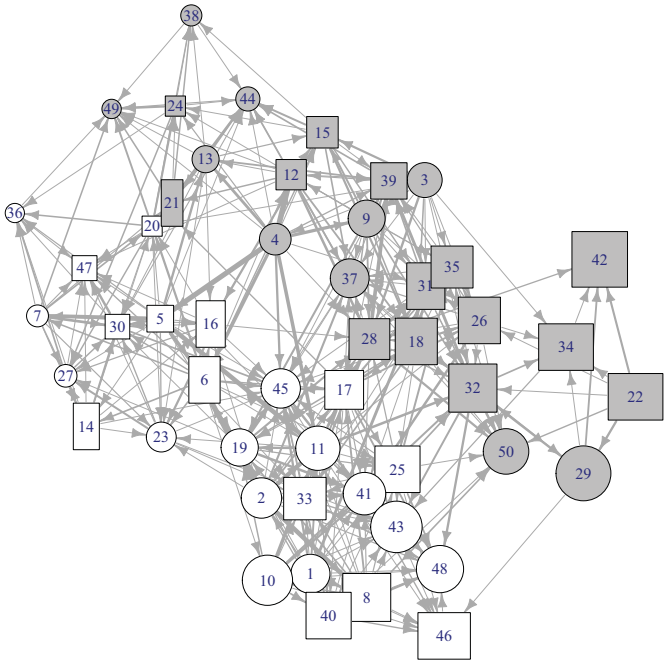


Figure 5.7: An Example of Temporal Inhomogeneity on a Random Subset of the Network. Larger Sized Transactions (Larger Width Edges) are Observed on a Random Subset of the Network.

Upon change detection, we rely of VI measures for diagnostics. Figures 5.9 and 5.10 show plots of VI at iteration 0 and K of the iterative forest algorithm. As depicted in the figures, the VI at iteration 0 does not provide effective diagnostics. The VI at the Kth iteration, on the other hand, correctly identifies the important attributes. This is again due to the partial nature of change that is detected and diagnosed through the iterations.

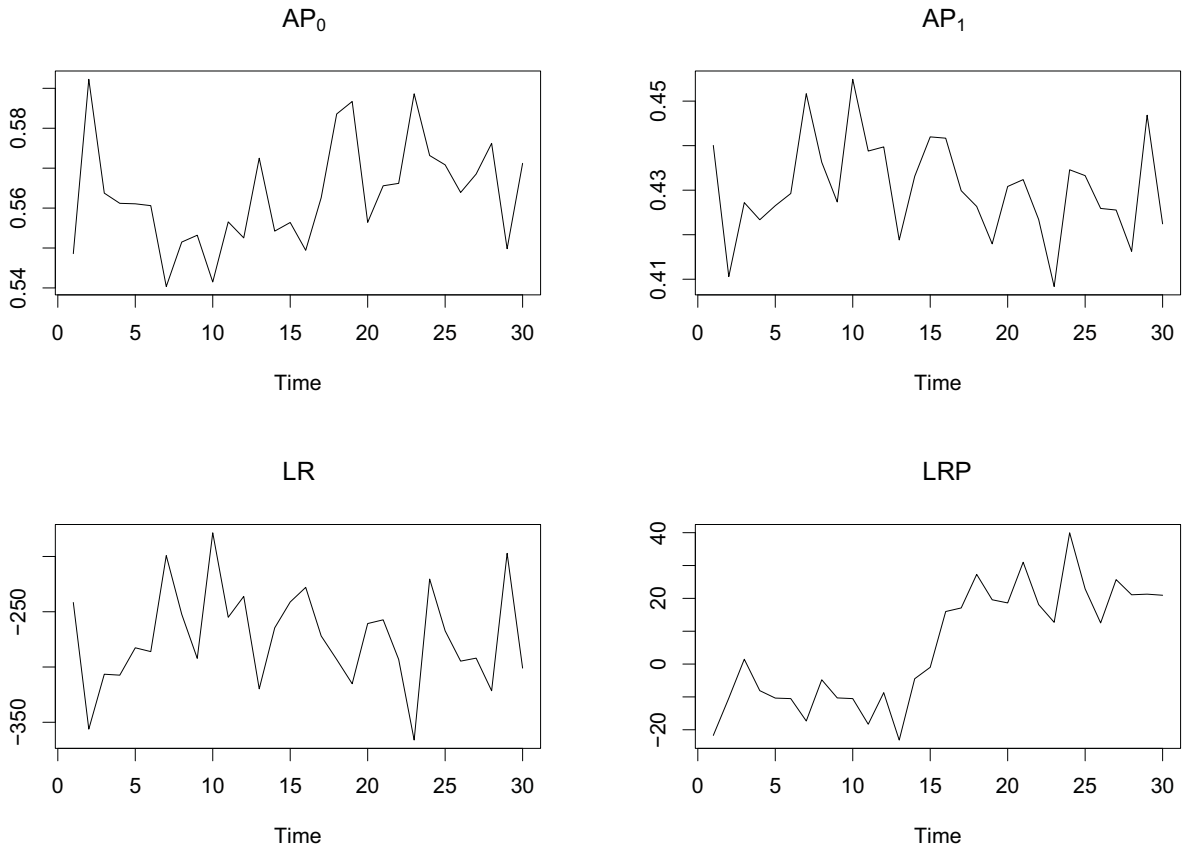


Figure 5.8: Plot of Different Monitoring Statistics Versus Time to Detect a Change in Transaction Size on a Small Random Subset of the Network. The Change is Clearly Detected by LRP that Considers Temporal Inhomogeneity on a Subset of Network.

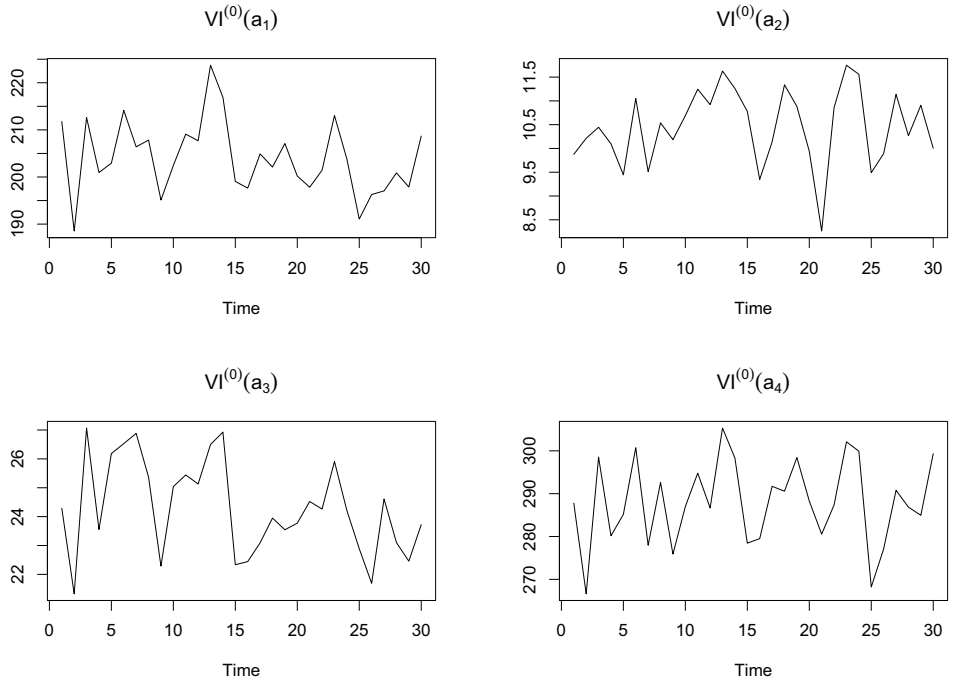


Figure 5.9: Plot of Variable Importance at Iteration 0 of the Iterative Forest Algorithm ($VI^{(0)}$) Versus Time. A Change is Induced at $t = 15$ that Distorts the Transaction Size on a Random Subset of the Network and is Detected Through the Monitoring Statistics. The Nature of the Change is Not Identified at the First Iteration.

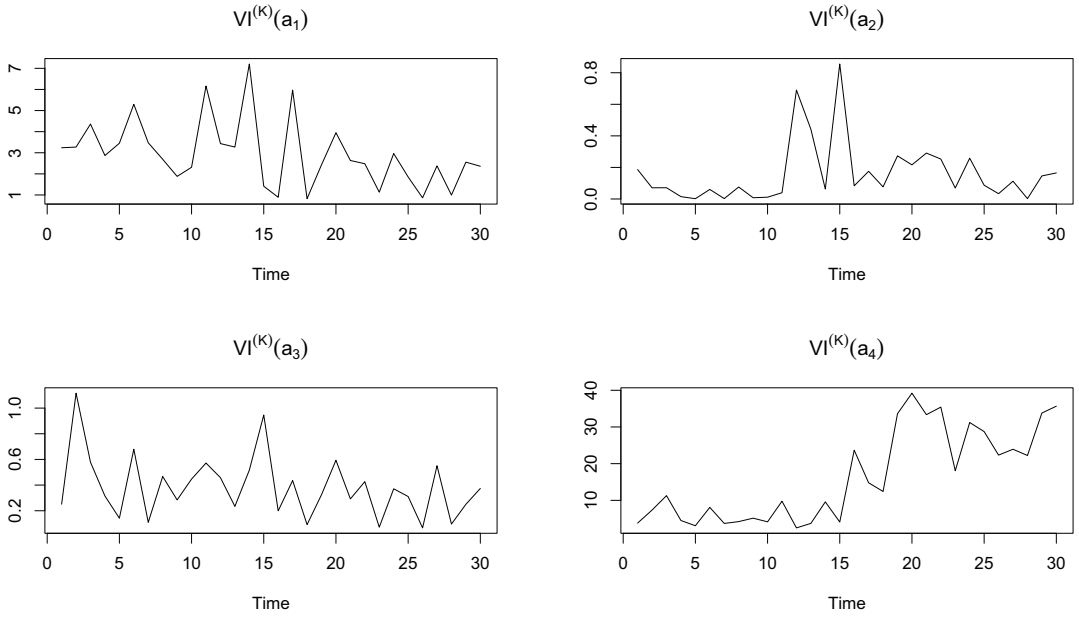


Figure 5.10: Plot Of Variable Importance at Iteration K (Last Iteration) of the Iterative Forest Algorithm ($VI^{(K)}$) Versus Time. A Change is Induced at $t = 15$ that Distorts the Transaction Size on a Random Subset of the Network and is Detected Through the Monitoring Statistics. The Increase in the $VI^{(K)}(a_4)$ Correctly Identifies the Nature of the Change.

5.3.2 Networks With No Vertex And Edge Attributes

Although the focus of this chapter is monitoring networks with vertex and edge attributes, this section demonstrates the proposed method's applicability for monitoring networks where no vertex or edge attributes are available. In such cases, we have access to only the topological structure (vertices and edges) of the network at each time stamp. We consider the detection of a "chatter" anomaly (Park *et al.*, 2013) where a small, unspecified subset of the vertices are involved in excessive communication during some time period.

Network generation scheme: We use the generation scheme used in Park *et al.* (2013) that is a modified Erdos-Renyie model. The network consists of V vertices, r of which are involved in the chatter. An edge between two vertices is modeled

as Bernoulli random variable. The probability of success is q for pairs involved in the chatter and $p, p < q$ otherwise. We use $ER(V, p, q, r)$ to refer to this generation scheme. In the experiments $V = 50$, $r = 6$, $p = 0.01$ and $q = \{0.2, 0.3, 0.4, 0.5\}$ are used.

In the absence of vertex and edge attributes, we rely solely on the topological attributes. We use the vertices degrees, the Jaccard and Dice similarity coefficient as the attributes (Adamic and Adar, 2003). That is, each transaction is defined as a four dimensional vector (a_1, a_2, a_3, a_4) where a_1 is the degree of the origin vertex, a_2 is the degree of the destination vertex, a_3 is the Jaccard coefficient and a_4 is the Dice coefficient.

Following the experiment in Park *et al.* (2013), the control limit is established based on fixing the type I error to be 0.05. The LRP statistic is then compared to the proposed method in Park *et al.* (2013) based on detection power on 500 networks. This method is monitoring a statistic that is a linear combination (referred to as fusion) of network measures (including the scan static over subgraphs on the network), which we denote by FGI. Part (a) of Figure 5.11 summarizes the results and reflects the superiority of the LRP statistic compared to FGI.

In comparing the proposed method to other methods, besides the detection power, two other important issues need to be considered. First is the applicability of the method for monitoring networks with attributes (e.g., vertex and edge attributes). Monitoring approaches limited to the network topology ignore this important data and, therefore, are not applicable for many monitoring applications. The proposed method, however, is suitable for monitoring when additional attributes are available. The proposed method in Park *et al.* (2013) is, however, limited to the network topology. The second is the scaling of the method for large networks. Most methods that monitor network measures extract the measures by enumerating windows (e.g. sub-

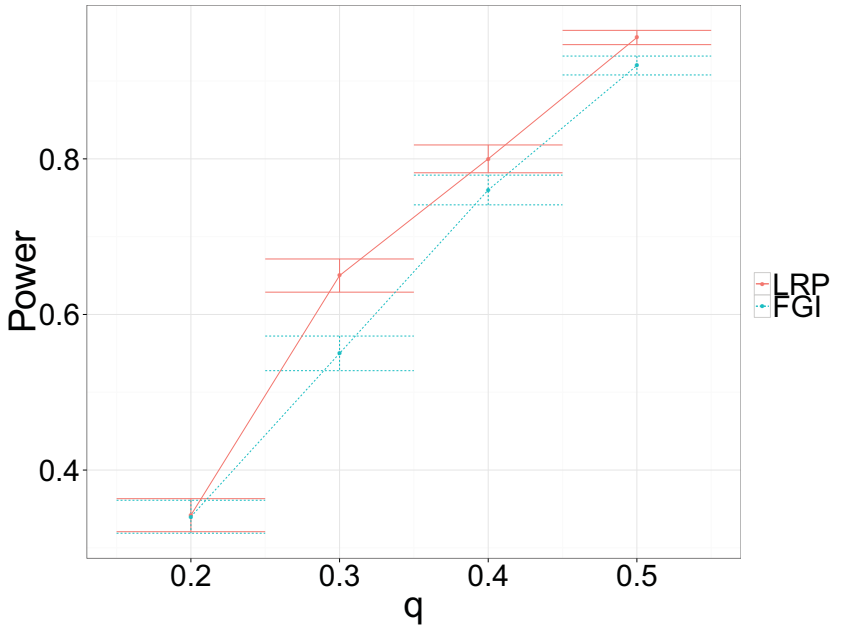
graphs) over the entire network. The time complexity of enumeration over networks is at least quadratic (Kiyomi, 2006) in the number of vertices. The method proposed in this chapter, however, is loglinear in the number of edges and, hence, scales to networks of large size.

We next investigate the performance of the proposed method for monitoring large networks. In this direction, networks are generated according to a modified $ER(V, p, q, r)$. Specifically, the number of edges between two vertices is modeled as a Binomial random variable with m trials. We refer to this generation scheme as $ER(V, p, q, r, m)$. This simple modification allows for controlling the size of the network. We use $V = 50$, $p = 0.1$, $q = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and $m = \{5, 10\}$. The additional trials and higher p and q values result in networks with larger number of transactions. Part (b) of Figure 5.11 demonstrates the increase in the power of the LRP statistic for larger networks.

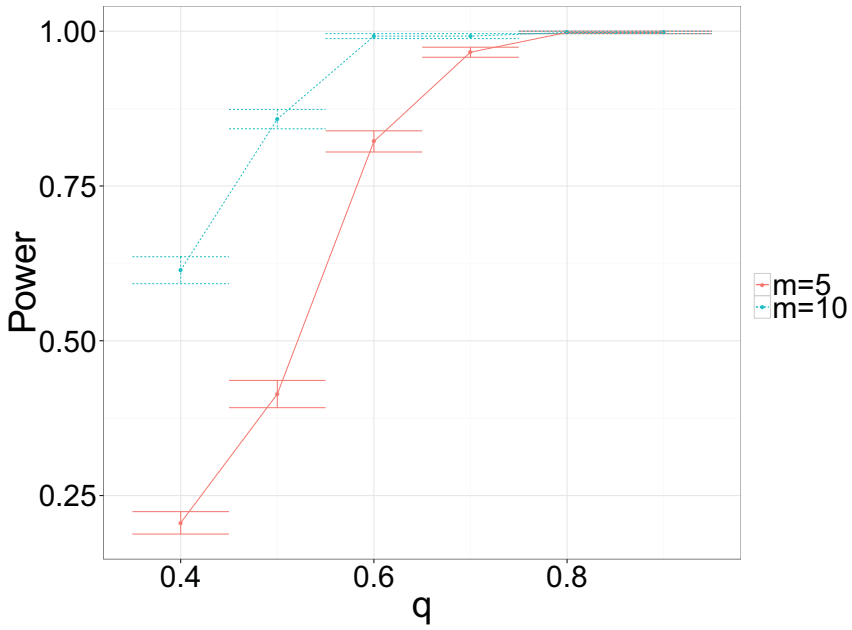
5.3.3 The Enron Email Network

The application of the proposed method for monitoring a dynamic network from the Enron corpus (Priebe *et al.*, 2005) is demonstrated in this section. The data consists of email communications between Enron employees from 1998 to 2002. This is modeled as a stream of directed networks. Each employee is represented as a vertex and an edge between two vertices indicates at least one email (on the same topic) sent between the pair in a one week time interval. Each vertex also has an attribute that denotes the role of the employee represented by the vertex. Possible roles are "President", "Director", "Trader", "CEO" and "Other". Two categorical attributes a_1 and a_2 summarizing the sender and receiver's roles are created. Also, the topic of the emails, provided by Berry *et al.* (2001) is used as an edge attribute a_3.

Towards monitoring the email communication, the weekly email communications



(a)



(b)

Figure 5.11: Plot of Detection Power Versus Different q Values. Part (a) Shows the Comparison of the LRP and FGI Statistic for $ER(V = 50, p = 0.01, q = \{0.2, 0.3, 0.4, 0.5\}, r = 6)$. The Superiority of the LRP is Evident. Part (b) Shows the Effect of Network Size on The Detection Power of the LRP Statistic for $ER(V = 50, p = 0.1, q = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}, r = 6, m = \{5, 10\})$. An Increase in Power for Larger Networks is Evident.

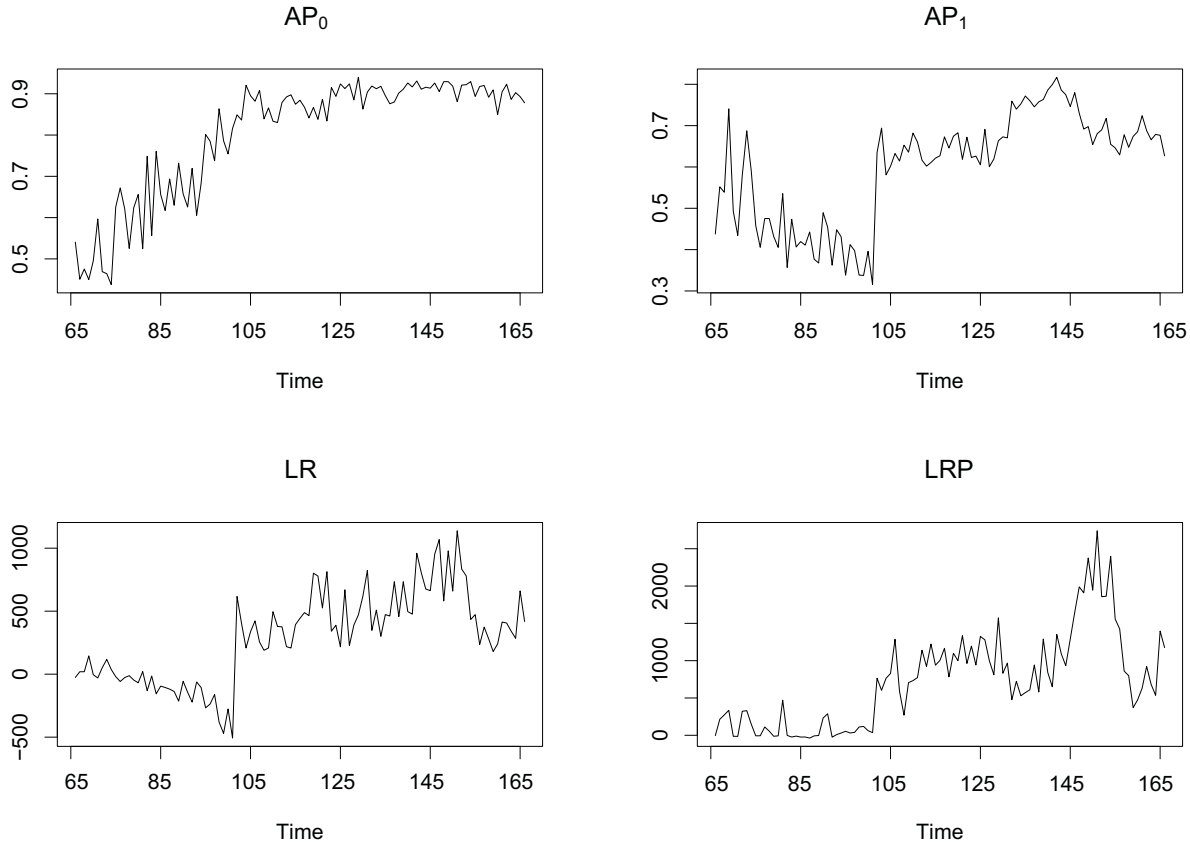


Figure 5.12: Plot of Monitoring Statistic Versus Time for the Enron Network. The Weekly Email Communications is Compared to the Email Communication In A Reference Month (The 55th Week To 65th Week Of 1998). The Monitoring Reveals Different Levels Of Temporal Inhomogeneity Through Time.

is compared to the email communication over a reference month (the 55th week to 65th week of 1998). Prior data is disregarded due to the scarce email communication. We apply monitoring statistic AP_0, AP_1, LR and LRP to this data which depicts the presence of different levels of temporal inhomogeneity through time (refer to Figure 5.12). The $VI^{(K)}$ measures are examined in Figure 5.13 that provide several interesting insights about the nature of the temporal inhomogeneities such as changes in the email topics.

In the absence of knowledge about the ground truth of the temporal inhomogeneity of this real multi-dimensional network, we rely on some visualization to gain further

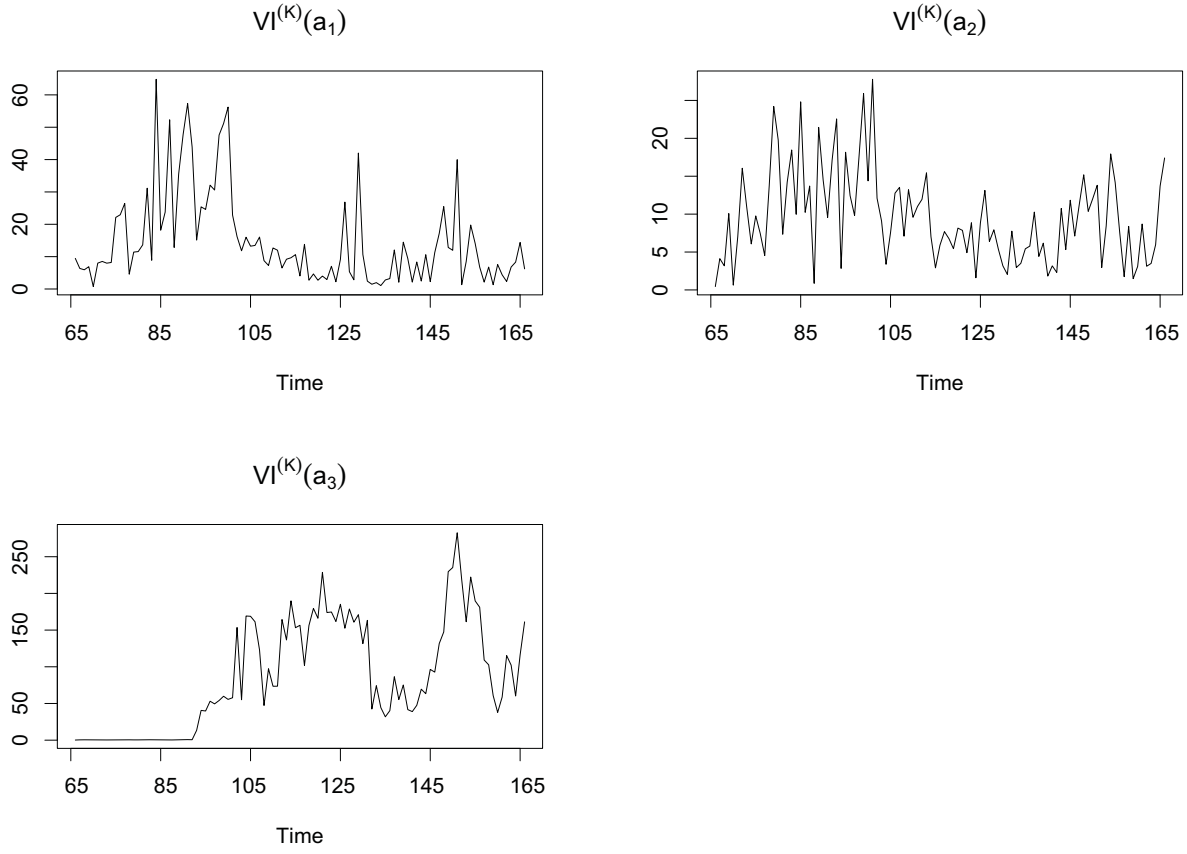


Figure 5.13: Plot Of $VI^{(K)}$ Measures Versus Time for the Enron Data Providing Insight on the Nature of Temporal Inhomogeneity.

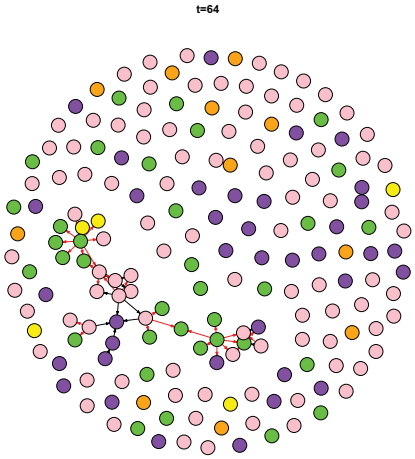
insight on the results. We note that a full interpretation of the result is beyond the scope of this chapter and provide interpretation on only a simple finding of the method. Figure 5.14 and 5.15 depict the email communication of four different weeks. Each employee is depicted as a vertex, colored with respect to role and each edge depicts at least one email (of the same topic) between the pair and is colored with respect to the email's topic. The networks in parts (a) and (b) of Figure 5.14 pertain to networks in the reference set. Note that the unconnected vertices are the vertices that were not involved in any email communication in that particular week. The network in part (a) of Figure 5.15 depicts a network where the monitoring statistics (LR and LRP for example) depict a modest value (in comparison to the rest, this

is week 75). Finally the network in part (b) pertains to the week with the highest value of the LRP statistic and high value for LR (week 151). An insight from the visualization is the higher variety of email topics during week 151 compared to the other weeks depicted (different colored edges). This is reflected by the peak in the $VI^{(K)}(a_3)$ during this week.

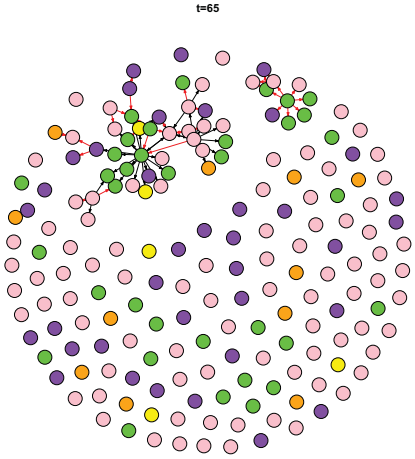
5.3.4 Detection of Partial Inhomogeneity

In this section, we study the sensitivity of the LRP statistic for detecting partial temporal inhomogeneity. The experiment is based on networks with 1000 vertices. Each vertex is associated with three vertex attributes α_1, α_2 and α_3. Additionally, each edge is associated with a size described through attribute a_4. The distributions of the attributes are the same as the attributes discussed in the network generation scheme of Section 5.3.1 so that a transaction is defined by a four dimensional vector (a_1, a_2, a_3, a_4), where a_1, a_2 and a_3 are defined by Equations 5.16, 5.17 and 5.18. We consider changes to the a_4 attribute. Under no change $a_4 \sim Normal(1, 1)$ (as in Section 5.3.1). Under change the distribution shifts to $Normal(\mu + \delta\sigma, 1)$. Changes of different magnitudes (different δ values) are considered. Also, we let T denote the average number of transactions in each network and consider networks with different T values. Finally, we let U denote the number of transactions per network whose size (attribute a_4) follows the Normal distribution with shifted mean and consider different values for U. Table 5.1 summarizes the experimental settings.

A change after time $t = 20$ is imposed according to the experimental settings of Table 5.1 to a_4. Experiments indicate the poor detection of the AP_0, AP_1, LR statistic. The LRP statistic, on the other hand, has better detection. We present the results for the LRP in Figures 5.16, 5.17, 5.18. As depicted, changes with shifts of larger magnitude that involve larger percentage of transactions in larger networks

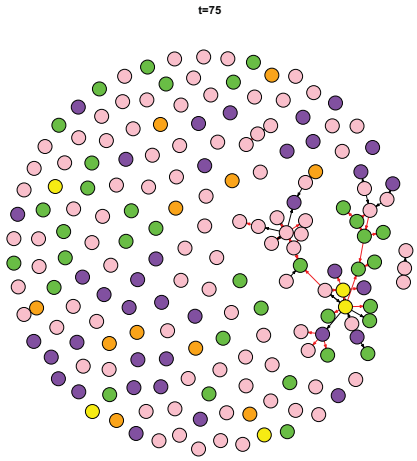


(a)

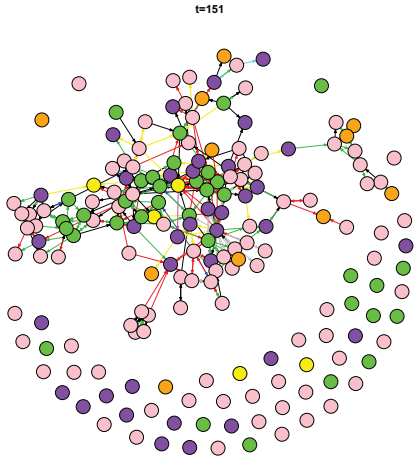


(b)

Figure 5.14: Enron's Email Network at Different Weeks. The Networks in Parts (a) and (b) Pertain to Networks in the Reference Set.



(a)



(b)

Figure 5.15: Enron's Email Network at Different Weeks. The Network in Part (a) Depicts a Network where the Monitoring Statistics Depict a Modest Value (Week 75). Finally the Network in Part (b) Pertains to the Time Stamp with the Highest Value of the LRP Statistic (week 151).

Parameter	Value
Magnitude of shift(δ)	3, 5, 7
Average number of transactions(T)	2000, 7000
Percentage of transactions under change(U)	1, 2, 5

Table 5.1: Experimental Settings.

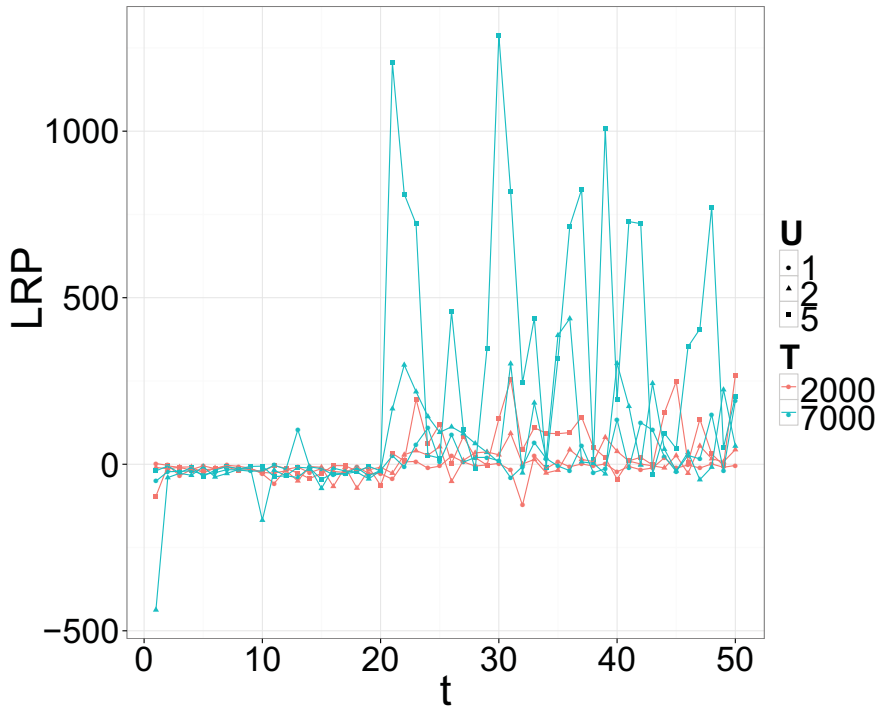


Figure 5.16: Plots of the LRP Monitoring Statistic Versus Time. Change is Imposed After $t = 20$ According to the Experimental Settings in Table 5.1 and is Detected Through the Monitoring. Cases for $\delta = 3$ are Shown.

are detected easier.

We next extend the above experiments to consider monitoring network with high dimensional transactions. Specifically, the networks are composed of 100 dimensional transactions $(a_1, a_2, a_3, a_4, a_5, \cdots, a_{100})$. Attributes $a_1 - a_4$ are defined as above, attributes $a_5 - a_8$ are degree of the origin, destination, Jaccard and Dice respectively and attributes $a_9 - a_{100}$ are $Normal(1, 1)$. Change is imposed on a_4 similar to the

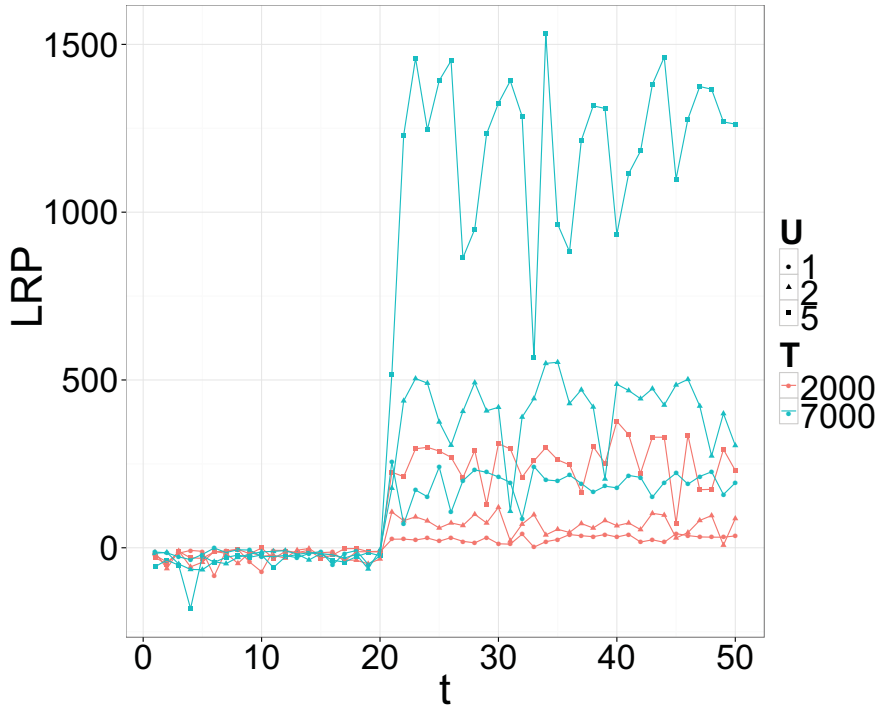


Figure 5.17: Plots of the LRP Monitoring Statistic Versus Time. Change is Imposed After $t = 20$ According to the Experimental Settings in Table 5.1 and is Detected Through the Monitoring. Cases for $\delta = 5$ are Shown.

above experiments with $\delta = 7$. Results are shown in Figure 5.19. It should be noted that the change imposed in this later experiment is extremely subtle: it affects only a few percent of the transactions $(1, 2, 5\%)$ on 1% of the attributes.

Results convey high variability of the LRP statistic before change is imposed. Possibly, this is due to the greedy nature of the algorithm that assigns transactions a class 1 if $\hat{p}_1(e_i) > 0.5$. Results may improve if slow learning where class assignment is randomized based on the current $\hat{p}_1(e_i)$ values is implemented.

As a final note, it should be mentioned that the approach presented here is sensitive to changes that are captured in the distribution of the attributes. However, this might require the attributes to be enhanced in some cases. For example, consider monitoring a stream of unattributed networks where each vertex has an associated ID. The reference set is characterized by the $ER(V; p; q; r)$ model. That is, each net-

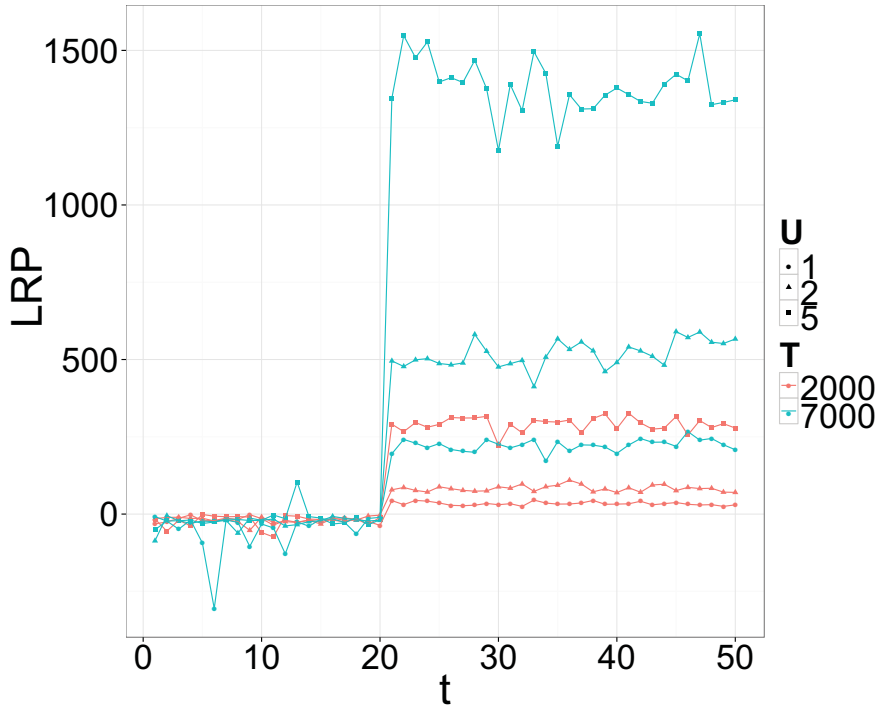


Figure 5.18: Plots of the LRP Monitoring Statistic Versus Time. Change is Imposed After $t = 20$ According to the Experimental Settings in Table 5.1 and is Detected Through the Monitoring. Cases for $\delta = 7$ are Shown.

work consists of V vertices, r of which are involved in the chatter and the probability of an edge is q for pairs involved in the chatter and p otherwise, where $p < q$. Let s_1 denote the set of vertices that are involved in the chatter. Now, consider a change that is also characterized by the $ER(V; p; q; r)$ model but with a different set of vertices involved in the chatter. Let s_2 denote this set, where $s_1 \neq s_2$ and $|s_1| = |s_2| = r$. Following the monitoring approach outlined in Section 5.3.2, each transaction may be defined as a four dimensional vector (a_1, a_2, a_3, a_4) where a_1 is the degree of the origin vertex, a_2 is the degree of the destination vertex, a_3 is the Jaccard coefficient and a_4 is the Dice coefficient. Contrasting transactions with these attributes from the reference set and the changed network fails to detect the change as the change is not captured in the distribution of the attributes.

The change discussed in the previous paragraph is better captured through incor-

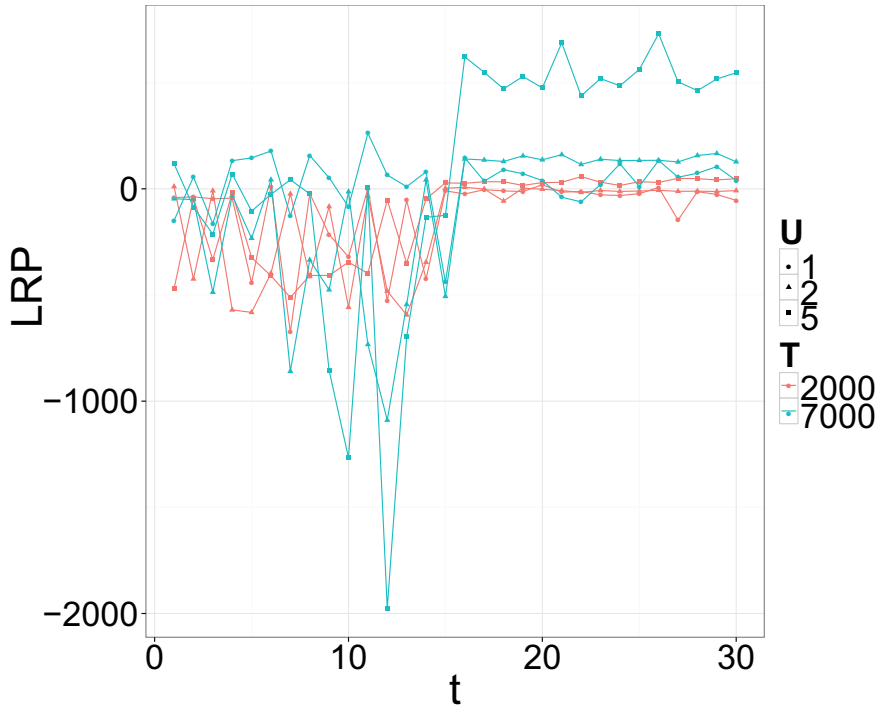


Figure 5.19: Plots of the LRP Monitoring Statistic Versus Time. Change is Imposed After $t = 15$ but Now Transactions Include 100 Attributes. Cases with $\delta = 7$ are Shown. It Should be Noted that the Change Imposed is Extremely Subtle as it Affects Only a Few Percent of the Transactions $(1, 2, 5\%)$ on 1% of the Attributes.

porating additional vertex-specific attributes in the transaction. In particular, the attributes may be enhanced to include the vertex IDs so that each transaction is defined as a six dimensional vector $(a_1, a_2, a_3, a_4, a_5, a_6)$, where a_5 and a_6 are the origin's and destination's vertex IDs and a_1 through a_4 are defined as before. Applying the proposed method to transactions with the additional attributes allows us to detect the mentioned change. This is because the distribution of attributes is now sensitive to the change.

5.4 Conclusion

The dynamics of entity interactions in complex, real world systems generate network streams. Further complexity is introduced through the layers of data provided

by vertex and edge attributes. This chapter studies monitoring the multiple facets of transactions on such multi-dimensional networks. Two important issues arise in this context. First is the detection of change in any region defined by transactions' high dimension feature space. The second is the detection of change that only affects a small subset of the transactions (referred to as partial temporal inhomogeneity). A monitoring method that addresses these two important issues is proposed.

By transforming the network monitoring problem to one of supervised learning, the proposed method leverages additional byproducts provided by many learners towards monitoring. The class probability estimates are used towards deriving novel monitoring statistics and the variable importance scores are used as diagnostics tools for insight on the temporal inhomogeneity.

Chapter 6

CONCLUSION

This dissertation introduces holistic learning as the integration of a comprehensive set of relationships that are used towards the learning objective. Specifically the focus is on multi-target and network monitoring problems for which a set of holistic learning algorithms are developed.

Chapter 3 introduces a novel tree-based ensemble method called the compound forest (CF) for the multi-target problem that leverages the relationships across multiple target attributes towards improving prediction accuracy. The embedding of the relationships in the learning algorithm allows for improved prediction performance in the presence of useful relationships while remaining robust in their absence. The method is justified through its connections to existing methods such as output smearing (Breiman, 2000), adaptive nearest neighbor (Hastie and Tibshirani, 1996) and importance sampled learning ensemble (Friedman and Popescu, 2003). In addition, experimental evaluation provides significant evidence for the benefit of CF, with the biggest improvements resulting from training the trees on a large number of relevant target attributes. Furthermore, the experiments depict the robustness of the method in the presence of a large number of target attributes that are of low relevance to each other. We show the versatility of CF in handling these characteristics on synthetic and real data from different domains.

For future research, a clustering scheme to group similar target attributes may be pursued. The distance measure used in clustering may be a function of the node impurity of the partitions of the feature space obtained using one target attribute with respect to the other target attributes. Also, in the current implementation of CF,

the assignment of weights is done through solving a regularized regression problem that takes other base learners into account. However, since the base learners are constructed in a parallel fashion, they are constructed independently without taking into account the other base learners. Future work can include the development of a serial approach that takes the previous constructed base learners into account.

Chapters 4 and 5 focus on networks that present a rich set of attributes and relationships for which holistic learning is important. Specifically, monitoring such systems through a holistic view that takes into account the relationships of multiple networks attributes is studied. The focus is on the difficult task of detecting a change in only a subregion of a high-dimensional space of network attributes that requires an integrated, holistic learning approach. Two monitoring algorithms are developed. The first method leverages vertex attributes in modeling and monitoring networks through a logistic regression framework. The second method extends the first to include vertex, edge and topological attributes in network modeling. This method transforms the monitoring task into an expedient structure for a machine learning algorithm. The transformation provides a powerful set of tools for addressing many important issues. These include the detection of changes that may only be local to subregions of a high-dimensional space of network attributes, the detection of changes that may impact only a small subset of the network and finally the development of diagnostic tools that shed light on the nature of change. Experimental evaluation depicts the heightened sensitivity of monitoring algorithms that embed network attributes in monitoring for detecting local change in subregions. Moreover, the benefits of a monitoring statistic that is specially tailored to detect changes that impact only a small subset of the network is shown. The statistic is based on a non-parametric estimation algorithm. For future work, we will investigate the application of this algorithm to clustering and non-network related statistical process control applications.

Adamic, L. A. and E. Adar, "Friends and Neighbors on the Web", Social networks **25**, 3, 211–230 (2003).

Aho, T., B. Zenko, S. Dzeroski and T. Elomaa, "Multi-Target Regression with Rule Ensembles", Journal of Machine Learning Research **1**, 1–48 (2012).

Al Hasan, M., V. Chaoji, S. Salem and M. Zaki, "Link Prediction using Supervised Learning", in "SDM Workshop on Link Analysis, Counter-terrorism and Security", (2006).

Azarnoush, B., J. Bekki, B. Bernstein and G. C. Runger, "An Associative Based Approach To Analyzing An Online Learning Environment", in "ASEE", (2013).

Barabási, A. and R. Albert, "Emergence of Scaling in Random Networks", Science **286**, 5439, 509–512 (1999).

Berry, M. W., M. Browne and B. Signer, "Topic Annotated Enron Email Data Set", in "Linguistic Data Consortium", (2001).

Blockeel, H., L. De Raedt and J. Ramon, "Top-Down Induction of Clustering Trees", in "Proceedings of the 15th International Conference on Machine Learning", pp. 55–63 (1998).

Breiman, L., "Bagging Predictors", Machine learning **24**, 2, 123–140 (1996).

Breiman, L., "Randomizing Outputs to Increase Prediction Accuracy", Machine Learning **40**, 3, 229–242 (2000).

Breiman, L., "Random Forests", Machine learning **45**, 1, 5–32 (2001).

Breiman, L. and J. H. Friedman, "Predicting Multivariate Responses in Multiple Linear Regression", Journal of the Royal Statistical Society: Series B (Statistical Methodology) **59**, 1, 3–54 (2002).

Breiman, L., J. H. Friedman, C. J. Stone and R. Olshen, *Classification and Regression Trees* (Chapman & Hall/CRC, 1984).

Capizzi, G. and G. Masarotto, "Self-Starting CUSCORE Control Charts for Individual Multivariate Observations", Journal of Quality Technology **42**, 2, 136–151 (2010).

Caruana, R., "Multitask Learning: A Knowledge-Based Source of Inductive Bias", in "Proceedings of the Tenth International Conference on Machine Learning", pp. 41–48 (1993).

Caruana, R., "Multitask Learning", Machine learning **28**, 1, 41–75 (1997).

Caruana, R., *Multitask Learning* (Springer, 1998).

Casella, G. and R. L. Berger, *Statistical Inference*, vol. 70 (Duxbury Press Belmont, CA, 1990).

Chakrabarti, D., Y. Zhan and C. Faloutsos, "R-MAT: A Recursive Model for Graph Mining", in "SDM", vol. 4, pp. 442–446 (SIAM, 2004).

Dávila, S., G. Runger and E. Tuv, "High-Dimensional Surveillance", in "ICANN 2011", pp. 245–252 (2011).

De'Ath, G., "Multivariate Regression Trees: a New Technique for Modeling Species-Environment Relationships", Ecology **83**, 4, 1105–1117 (2002).

Dempster, A. P., N. M. Laird, D. B. Rubin *et al.*, "Maximum Likelihood from Incomplete Data via the EM Algorithm", Journal of the Royal statistical Society **39**, 1, 1–38 (1977).

Demšar, D., S. Džeroski, T. Larsen, J. Struyf, J. Axelsen, M. B. Pedersen and P. H. Krogh, "Using Multi-Objective Classification to Model Communities of Soil Microarthropods", Ecological Modelling **191**, 1, 131–143 (2006).

Deng, H., G. Runger and E. Tuv, "System Monitoring with Real-Time Contrasts", Journal of Quality Technology **44**, 1, 9–27 (2012).

Erdos, P. and A. Renyi, "On Random Graphs I", Publicationes Mathematicae Debrecen **6**, 290–297 (1959).

Esposito Vinzi, V., C. M. Ringle, S. Squillacciotti and L. Trinchera, "Capturing and Treating Unobserved Heterogeneity by Response Based Segmentation in PLS Path Modeling", Tech. rep., ESSEC Research Center, ESSEC Business School (2007).

Fan, J., C. Zhang and J. Zhang, "Generalized likelihood ratio statistics and wilks phenomenon", Annals of statistics pp. 153–193 (2001).

Frank, O. and D. Strauss, "Markov Graphs", Journal of the American Statistical Association **81**, 395, 832–842 (1986).

Freeman, L., "Centrality in Social Networks Conceptual Clarification", Social Networks **1**, 3, 215–239 (1979).

Freeman, L. C., "A Set of Measures of Centrality Based on Betweenness", Sociometry **40**, 35–41 (1977).

Friedman, J. H., "Greedy Function Approximation: a Gradient Boosting Machine", Annals of Statistics **29**, 5, 1189–1232 (2001).

Friedman, J. H., T. Hastie and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics, 2001).

Friedman, J. H., T. Hastie and R. Tibshirani, "A Note on the Group Lasso and a Sparse Group Lasso", Tech. rep., Department of Statistics, Stanford University (2010).

Friedman, J. H. and B. E. Popescu, "Importance Sampled Learning Ensembles", Journal of Machine Learning Research **94305**, 1–32 (2003).

Gong, N. Z., A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov and D. Song, "Jointly Predicting Links and Inferring Attributes using a Social-Attribute Network", in "Social Network Analysis-Knowledge Discovery & Data Mining", (2011).

Hanneke, S., W. Fu and E. P. Xing, "Discrete Temporal Models of Social Networks", Electronic Journal of Statistics **4**, 585–605 (2010).

Hastie, T. and R. Tibshirani, "Discriminant Adaptive Nearest Neighbor Classification", Pattern Analysis and Machine Intelligence, IEEE Transactions on **18**, 6, 607–616 (1996).

Ho, Q., L. Song and E. P. Xing, "Evolving Cluster Mixed-Membership Blockmodel for Time-Evolving Networks", in "International Conference on Artificial Intelligence and Statistics", pp. 342–350 (2011).

Ho, T. K., "The Random Subspace Method for Constructing Decision Forests", Pattern Analysis and Machine Intelligence, IEEE Transactions on **20**, 8, 832–844 (1998).

Hoff, P. D., A. E. Raftery and M. S. Handcock, "Latent Space Approaches to Social Network Analysis", Journal of the American Statistical Association **97**, 460, 1090–1098 (2002).

Hosmer Jr, D. W. and S. Lemeshow, *Applied Logistic Regression* (John Wiley & Sons, 2004).

Kim, M. and J. Leskovec, "Multiplicative Attribute Graph Model of Real-World Networks", in "Algorithms and Models for the Web-Graph", pp. 62–73 (2010).

Kiyomi, M., *Studies on Subgraph and Supergraph Enumeration Algorithms*, Ph.D. thesis, PhD thesis, Department of Informatics, The Graduate University for Advanced Studies, Japan (2006).

Kocev, D., C. Vens, J. Struyf and S. Džeroski, "Ensembles of Multi-Objective Decision Trees", in "Proceedings of the 18th European Conference on Machine Learning", vol. 4701, pp. 624–631 (2007).

Kocev, D., C. Vens, J. Struyf and S. Džeroski, "Tree Ensembles for Predicting Structured Outputs", Pattern Recognition **46**, 3, 817–833 (2013).

Kumar, R., J. Novak, P. Raghavan and A. Tomkins, "Structure and Evolution of Blogspace", Communications of the Association for Computing Machinery **12**, 47, 35–39 (2004).

Leskovec, J., J. Kleinberg and C. Faloutsos, "Graph Evolution: Densification and Shrinking Diameters", Association of Computing Machinery Transactions on Knowledge Discovery from Data **1**, 1, 2 (2007).

Li, J., J. Jin and J. Shi, "Causation-Based T^2 Decomposition for Multivariate Process Monitoring and Diagnosis", Journal of Quality Technology **40**, 1, 46–58 (2008).

Lin, Y. and Y. Jeon, "Random Forests and Adaptive Nearest Neighbors", Journal of the American Statistical Association **101**, 474, 578–590 (2006).

Maboudou-Tchao, E. M. and D. M. Hawkins, "Self-Starting Multivariate Control Charts for Location and Scale", Journal of Quality Technology **43**, 2, 113–126 (2011).

MacGregor, J. F., C. Jaeckle, C. Kiparissides and M. Koutoudi, "Process Monitoring and Diagnosis by Multiblock PLS Methods", AIChE Journal **40**, 5, 826–838 (1994).

Marchette, D., "Scan Statistics on Graphs", Wiley Interdisciplinary Reviews: Computational Statistics **4**, 5, 466–473 (2012).

McCulloh, I. and K. M. Carley, "Detecting Change in Longitudinal Social Networks", Journal of Social Structure **12**, 3, 1–37 (2011).

Miller, B. A., N. Arcolano and N. T. Bliss, "Efficient Anomaly Detection in Dynamic, Attributed Graphs", in "IEEE International Conference on Intelligence and Security Informatics", pp. 179–184 (2013).

Montgomery, D. C., *Introduction to Statistical Quality Control*, vol. 2 (Wiley New York, 1991).

Myers, R. H., D. C. Montgomery, G. G. Vining and T. J. Robinson, *Generalized Linear Models: with Applications in Engineering and the Sciences*, vol. 791 (John Wiley & Sons, 2012).

Neil, J., C. Storlie, C. Hash, A. Brugh and M. Fisk, "Scan Statistics for the Online Detection of Locally Anomalous Subgraphs", Technometrics **55**, 4, 403–414 (2014).

Nesterov, Y., "Gradient Methods for Minimizing Composite Objective Function", Tech. rep., CORE (2007).

Pardo, L., *Statistical inference based on divergence measures* (CRC Press, 2005).

Park, Y., C. Priebe and A. Youssef, "Anomaly Detection in Time Series of Graphs using Fusion of Graph Invariants", IEEE Journal of Selected Topics in Signal Processing **7**, 1, 67–75 (2013).

Paynabar, K., J. Jionghua and A. B. Yeh, "Phase I Risk-Adjusted Control Charts for Monitoring Surgical Performance by Considering Categorical Covariates", Journal of Quality Technology **44**, 1, 39–53 (2012).

Pennock, D. M., G. W. Flake, S. Lawrence, E. J. Glover and C. L. Giles, "Winners Don't Take All: Characterizing the Competition for Links on the Web", Proceedings of the National Academy of Sciences **99**, 8, 5207–5211 (2002).

Perry, M. B., G. V. Michaelson and M. A. Ballard, "On the Statistical Detection of Clusters in Undirected Networks", Computational Statistics & Data Analysis **68**, 170–189 (2013).

Priebe, C. E., J. M. Conroy, D. J. Marchette and Y. Park, "Scan Statistics on Enron Graphs", Computational & Mathematical Organization Theory **11**, 3, 229–247 (2005).

Priebe, C. E., Y. Park, D. J. Marchette, J. M. Conroy, J. Grothendieck and A. L. Gorin, "Statistical Inference on Attributed Random Graphs: Fusion of Graph Features and Content", Computational Statistics & Data Analysis **54**, 7, 1766–1776 (2010).

Rokach, L., *Data Mining with Decision Trees: Theory and Applications*, vol. 69 (World scientific, 2008).

Runger, G. C., F. B. Alt and D. C. Montgomery, "Contributors to a Multivariate Statistical Process Control Chart Signal", Communications in Statistics–Theory and Methods **25**, 10, 2203–2213 (1996).

Sarkar, P. and A. W. Moore, "Dynamic Social Network Analysis using Latent Space Models", in "Special Interest Group on Knowledge Discovery & Data Mining Explorations", vol. 7, pp. 31–40 (2005).

Segal, M. and Y. Xiao, "Multivariate Random Forests", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **1**, 1, 80–87 (2011).

Simon, N., J. Friedman, T. Hastie and R. Tibshirani, "A Sparse-Group Lasso", Journal of Computational and Graphical Statistics, DOI **22**, 2, 231–245 (2013).

Snijders, T. A., "Models for Longitudinal Network Data", Models and Methods in Social Network Analysis **1**, 215–247 (2005).

Sullivan, J. H. and W. H. Woodall, "A Control Chart for Preliminary Analysis of Individual Observations", Journal of Quality Technology **28**, 3, 265–278 (1996).

Tatsuoka, M. M. and P. R. Lohnes, *Multivariate Analysis: Techniques for Educational and Psychological Research* . (Macmillan Publishing Co, Inc, 1988).

Thrun, S. and J. O'Sullivan, "Discovering Structure in Multiple Learning Tasks: The TC Algorithm", in "ICML", vol. 96, pp. 489–497 (1996).

Tibshirani, R., "Regression Shrinkage and Selection via the Lasso", Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996).

Tuddenham, R. D. and M. M. Snyder, "Physical Growth of California Boys and Girls from Birth to Eighteen Years.", Publications in child development. University of California, Berkeley **1**, 2, 183–364 (1953).

Wang, Y. J. and G. Y. Wong, "Stochastic Blockmodels for Directed Graphs", Journal of the American Statistical Association **82**, 397, 8–19 (1987).

Wasserman, S., *Social Network Analysis: Methods and Applications*, vol. 8 (Cambridge university press, 1994).

Witten, I. H. and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques* (Morgan Kaufmann, 2005).

Xu, K. S. and A. O. Hero III, "Dynamic Stochastic Blockmodels: Statistical Models for Time Evolving Networks", in "Social Computing, Behavioral-Cultural Modeling and Prediction", pp. 201–210 (2013).

Yuan, M. and Y. Lin, "Model Selection and Estimation in Regression with Grouped Variables", Journal of the Royal Statistical Society: Series B (Statistical Methodology) **68**, 1, 49–67 (2005).

Zhang, Y., N. Street and S. Burer, "Sharing Classifiers Among Ensembles from Related Problem Domains", in "Proceedings of the Fifth IEEE International Conference on Data Mining", pp. 522–529 (2005).