Network Kriging

Predicting the Attributes of Nodes in a Network

Daniel Hockey

A thesis submitted to the Faculty of Graduate and
Postdoctoral Affairs in partial fulfilment of the requirements
for the degree of

Master of Science

in

Probability and Statistics

Carleton University
Ottawa, Ontario

©2016, Daniel Hockey

Abstract

This thesis develops a method which predicts the role of a node in a social network. For illustrative purposes the network used is a subset of Al-Qaeda from 1998 which contains a total of 160 members.

While doing exploratory analysis on this network we noticed that there seemed to be an underlying connection with the distance between two members and their roles. This led to developing a prediction method which could exploit this correlation structure. We use the geostatistical prediction method called Kriging that is modified to perform in a network; which we call Network Kriging.

This thesis gives the background knowledge necessary to understand the techniques, shows the results of Network Kriging and compares results to those using the K-Nearest Neighbours algorithm. We found that for important roles, such as Emir (Leadership), Network Kriging performs better than K-Nearest Neighbours.

Acknowledgements

My student experience would not have been as rewarding without the support of both Dr. Shirley Mills and Dr. Song Cai. Both have provided me with an incredible amount of guidance not only with my school work, but also other issues that arise in a student's life. I am lucky to have them as mentors and will always be grateful for their hard work.

Contents

Abstract	i
Acknowledgements	ii
List of Tables	v
List of Figures	vi
1 Introduction to Social Networks	1
1.1 Basics	1
1.2 Centrality	3
1.2.1 Degree Centrality	3
1.2.2 Closeness Centrality	4
1.2.3 Betweenness Centrality	5
1.3 K-Nearest Neighbours (KNN)	7
1.4 Data	9
2 Kriging	11
2.1 Introduction	11
2.2 Stationarity	14
2.3 Variogram	14
2.3.1 Definition	14
2.3.2 Construction	15
2.3.3 Models and Properties	19
2.4 Finding the Weights in Kriging	20
2.5 Universal Kriging	22
3 Network Kriging	25
3.1 Network Kriging Method	25
3.2 Network Stationarity	26
3.2.1 Universal Network Kriging	27

3.2.2	Cluster Network Kriging	27
3.2.3	Neighbourhood Network Kriging	34
3.2.4	Remarks	34
3.3	Results	36
3.3.1	Emir (Leadership)	37
3.3.2	Finance/Logistics	42
3.3.3	Subordinate	46
3.3.4	Fatwa Committee	50
3.4	Remarks	53
4	Conclusion	58
	Bibliography	59

List of Tables

1	Confusion Matrix	8
---	----------------------------	---

List of Figures

1	Graph - Undirected	2
2	Adjacency Matrix - Undirected	2
3	Degree Centrality	4
4	Closeness Centrality	5
5	Betweenness Centrality	6
6	Betweenness Centrality in a larger network	6
7	Network in 1998	9
8	Heat Map Predictions	12
9	Kriging Example	12
10	Kriging Example - Weights	13
11	Variogram Cloud	16
12	Sample Variogram and Variogram Model	17
13	Network after clustering	33
14	1998 - Emir	37
15	Variogram - Emir	39
16	ROC 1 - Emir	40
17	ROC 2 - Emir	41
18	1998 - Finance/Logistics	42
19	Variogram - Finance/Logistics	43
20	ROC 1 - Finance/Logistics	44
21	ROC 2 - Finance/Logistics	45
22	1998 - Subordinate	46
23	Variogram - Subordinate	47
24	ROC 1 - Subordinate	48
25	ROC 2 - Subordinate	49
26	1998 - Fatwa Committee	50
27	Variogram - Fatwa Committee	51
28	ROC - Fatwa Committee	52
29	Subnetwork Variograms - Emir	54

30	Subnetwork Variograms - Finance/Logistics	55
31	Subnetwork Variograms - Subordinate	56
32	Subnetwork Variograms - Fatwa Committee	57

1 Introduction to Social Networks

We give a very brief overview of the way information is stored in networks and introduce common terminology. In particular we discuss the concept of centrality. The section also gives an explanation of K-Nearest Neighbours since it will be used as the benchmark for our method. An overview of the network used throughout the paper is given, including how the data was cleaned to suit our needs.

1.1 Basics

In social networks, like all networks, there are nodes and links (sometimes referred to as vertices and edges). Nodes are points of interest (e.g. people, places, objects). Links are the ways in which nodes connect (e.g. friendship, colleague, family). The strength of the connection between two nodes is called the weight of the link.

Analytical methods in social network analysis (SNA) use static networks. Subsequently if one has a dynamic network, SNA methods can only be used with static ‘snapshots’ of the network. Social networks also only have one type of node and one type of link. Analysis of the *evolution* of a network falls in the domain of dynamic network analysis (DNA) [3]. Social networks can be represented by graphs (Figure 1) and adjacency matrices (Figure 2) [1]. (Note the two figures below do not represent the same data.)

Figure 1: Graph - Undirected

	Node						
	1	2	3	4	5	6	7
1			1	1			
2	1			1		1	
3	1			1			
4	1	1	1			1	
5						1	
6		1		1	1		1
7						1	

Figure 2: Adjacency Matrix - Undirected

Networks can be either directed or undirected. A directed network has a specific direction in which information can flow between two nodes. In an undirected network the information always flows both ways between two connecting nodes. Figure 1 and Figure 2 are both examples of undirected networks. Directed networks are usually shown with directional arrows on the links. The arrows indicate the direction in which information flows. The adjacency matrix for a directed network will usually not be symmetric. Any directed network can be made undirected by simply making all directed links connect both ways, which in turn makes the adjacency matrix symmetric.

If the weights connecting the nodes are not binary then the network is called a valued network. One could imagine how identifying the weight of a link is important. There are various ways of classifying the weight of a link. The weight might be defined as frequency of communication. Or if one was looking at a disease outbreak, the link weight may represent the duration of exposure [6]. The distance between nodes is the reciprocal of the weight of the link. Therefore the higher the weight, the shorter the distance. The longest shortest path in the network is referred to as the diameter. In other words, out of all the shortest paths in the network, the diameter is the longest.

1.2 Centrality

A key concept in analyzing a network is centrality. Centrality uses properties of the network to characterize different nodes. The three most well-known types of centrality are: degree, closeness and betweenness.

1.2.1 Degree Centrality

Degree centrality is the number of links connected to a node; the more links the higher the degree centrality. Degree centrality can be mathematically defined as [7]

$$D(x_i) = \sum_{j=1}^n A_{ij},$$

where $D(x_i)$ is the degree centrality for node x_i , n is the total number of nodes, and A_{ij} is an entry in the adjacency matrix (refer to Figure 2) for the i^{th} row and j^{th} column. The node x_i is fixed and one adds the number of links connected to the node. In Figure 2, the degree centrality for $D(x_2)$ would be

$$D(x_2) = \sum_{j=1}^7 A_{2j} = 1 + 0 + 0 + 1 + 0 + 1 + 0 = 3.$$

Figure 3 shows degree centrality in a graphical representation. One can see that the nodes with larger radii have a higher degree centrality.

Figure 3: Degree Centrality

The explanation for Figure 3 is not very complicated. Clearly the node with the most links (6) has the largest radius.

There is also in-degree centrality and out-degree centrality, where in-degree centrality is the number of links coming in to a node and out-degree centrality is the number of links going out from a node. These are only used for directed networks.

1.2.2 Closeness Centrality

Closeness centrality looks at the distance of the link(s) that connect one node to another. This is usually a combination of links as it may take more than one link to connect two nodes. The sum of the distances from one node to another provides the total distance. Closeness centrality, by its name, measures overall how close a node x_i is to all other nodes. Therefore, it is defined as the reciprocal of the sum of such distances as follows [7]

$$C(x_i) = \frac{1}{\sum_{j=1}^n d_{x_i x_j}},$$

where $C(x_i)$ is the closeness centrality of node x_i and $d_{x_i x_j}$ is the shortest distance from node x_i to node x_j .

Figure 4: Closeness Centrality

In Figure 4, one can see that the four nodes in the middle of the network have the highest closeness centrality. This is because these nodes easily connect to the larger groups on the left and right hand sides of the network, thus making their paths to all possible nodes the shortest.

1.2.3 Betweenness Centrality

Betweenness centrality measures the ability of a node to connect other nodes in the network and hence measures how dependent the network is on the node for efficient communication [2]. Similar to closeness centrality, it is necessary to find the shortest path (shortest distance) between all nodes in the network. Then we observe how many times the node of interest lies on one of the shortest paths [7]. The betweenness centrality of node x_i is defined as

$$B(x_i) = \sum_{j \neq i}^n \sum_{k < j, k \neq i}^n \frac{m_{x_j x_k}(x_i)}{|m_{x_j x_k}|},$$

where $m_{x_j x_k}(x_i)$ is the number of times node x_i lies on one of the shortest paths between node x_j and node x_k , and $|m_{x_j x_k}|$ is the number of shortest paths between node x_j and node x_k (in case there is more than one shortest path with the same length) [7].

Figure 5: Betweenness Centrality

Betweenness centrality is a measure of how well the node helps the network flow. One might think that the two nodes in the middle of the network (identified by arrows in Figure 5) might have high betweenness centrality. However, this is not the case. If one of the two nodes were removed the network would still be able to communicate using the other. Betweenness centrality can be difficult to observe in simple networks. Figure 6 provides a much clearer picture.

Figure 6: Betweenness Centrality in a larger network

1.3 K-Nearest Neighbours (KNN)

The KNN algorithm is one of the most well-known methods for network prediction. If one is trying to predict attributes of a node, the KNN algorithm simply looks at whether or not the node's neighbours have the attribute. The KNN algorithm is defined as [9]

$$\hat{p}(y(x^*) = 1) = \frac{\sum_{i=1}^n y(x_i) \delta(x_i \in N)}{|N|},$$

where x^* is the node which we are trying to predict, and $y(x^*)$ is a binary variable with value 1 representing x^* possessing the attribute of interest, 0 otherwise. So the left side of the equality is the predicted probability that the value of the attribute for node x^* is equal to 1. In layman's terms, what is the estimated probability that node x^* possesses the attribute of interest?

The numerator of the right side of the equality is the number of nodes within a neighbourhood, N , that possess the attribute of interest. A neighbourhood is a subnetwork constructed using only the nodes that have a distance to the node of interest which is less than or equal to the neighbourhood size. Whether or not a node is in the neighbourhood of interest is expressed using Kronecker's delta where

$$\delta(x_i \in N) = \begin{cases} 1 & : x_i \in N, \\ 0 & : x_i \notin N. \end{cases}$$

The numerator is divided by the total number of nodes within the neighbourhood, $|N|$. The size of the neighbourhood is determined using cross-validation with a training set. How many neighbourhood sizes to try will be dependent on where the node of interest is located in the network and on the diameter of the network.

As was already stated, the value received from the KNN algorithm will

be a probabilistic value (a value between 0 and 1). If one wants to have a definitive estimation as to whether the node possesses that attribute, further work will need to be done. This will amount to finding a threshold τ (a cut off point), to determine if a predicted probability will score up to a 1 or down to a 0. This threshold is found by using cross-validation with a training set and a confusion matrix.

A threshold function works like this

$$\hat{y}(x_i) = \begin{cases} 1 & : \hat{p}(y(x^*) = 1) \geq \tau, \\ 0 & : \hat{p}(y(x^*) = 1) < \tau. \end{cases}$$

To find an appropriate threshold one can iterate through many different thresholds and find which is ‘optimal’. This is done by analyzing the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) rates at the different thresholds used. Where these rates are defined as,

$$\text{TP} = P(\hat{y}(x^*) = 1 | y(x^*) = 1)$$

$$\text{FP} = P(\hat{y}(x^*) = 1 | y(x^*) = 0)$$

$$\text{TN} = P(\hat{y}(x^*) = 0 | y(x^*) = 0)$$

$$\text{FN} = P(\hat{y}(x^*) = 0 | y(x^*) = 1).$$

One can easily visualize these rates using a confusion matrix. Below, Table 1 shows how the information is stored in the matrix.

	Predicted as 1	Predicted as 0
Actual value 1	TP	FN
Actual value 0	FP	TN

Table 1: Confusion Matrix

Ideally, one would want both the TP and TN rates to be 1. Unfor-

tunately, this is a rare circumstance. One then has to think about which of the TP, TN, FP, and FN rates are more important. This decision will vary based on the circumstance. Comparing techniques is rather difficult using confusion matrices as their values are subjective to the threshold used. Therefore the comparison of techniques later on will be evaluated using receiver operator characteristic curves (ROC curves).

In Eric Kolaczyk’s book *Statistical Analysis of Network Data - Methods and Models*, the KNN algorithm is explained using an example of predicting whether lawyers practise in litigation or corporate law. We will be using the KNN algorithm in later sections as a comparison to our technique.

1.4 Data

The example network used in this thesis is a dynamic network. More specifically, there are yearly snapshots of the network from 1998 to 2004. However, the methods we will be implementing are static methods, so we will not be looking at the network over time. For the purposes of this thesis we will be focusing on the data from 1998 as it contains the most nodes and links.

Figure 7: Network in 1998

The network is a subset of Al-Qaeda, a terrorist group. The nodes in the network are individuals associated with the group. Members in the network fulfil certain roles. These roles are: Emir (Leader), Military Committee, Fatwa Committee (Religious Leaders), Finance/Logistics, Media/Propaganda, Local Chief, and Subordinate.

The value of a role for a node in the network, $y(x_i)$, is a binary variable, therefore it is represented with a 1 or a 0. If the member has a value of 1 for a role then that is a role they fulfil, 0 otherwise. Our goal is to predict for a given role whether or not a member fulfils the role.

The network was originally not a connected network. All of the isolated nodes were removed since the prediction methods require connected networks. The links are directed, however for our purposes we will treat them as undirected for finding distances. The weights on the links are binary. This means that the distance between any two connecting nodes is 1, moreover any two nodes in the network will have some discrete distance from one another.

2 Kriging

Kriging is both the inspiration and the backbone of this thesis. This section gives an overview of Kriging and Universal Kriging and their assumptions. We will discuss how we adapt Kriging to suit our needs for prediction in a network in Section 3.

2.1 Introduction

Kriging is a spatial prediction method developed by Danie Krige [10]. It was originally developed to produce the best linearly unbiased predictor for a geophysical variable in a Euclidean space, which in application is a 2-D geographical region. It is called the “best” predictor because it is the predictor which minimizes squared error, this will be discussed in detail in Section 2.4.

To motivate Kriging we use the following example. Suppose one may want to predict the temperature or create a heat map for a plot of land even though there are no readings available at many locations. In this case Kriging will take the information from the locations where the temperature is known and then predict the temperature at the other regions.

Figure 8: Heat Map Predictions

In Figure 8 a heat map of the maximum temperature for an area has been created by using readings from the blue and green points. A method known as Universal Kriging was able to predict the temperatures at all the other locations on the grid. The prediction is done by analyzing the correlation structure between the points at different lags (distances). Below we have a simple example.

Figure 9: Kriging Example

Suppose we are trying to discover where oil might be located in the ground. We have done test drilling at all of the black points in Figure 9

and recorded some value as to how much oil was there. Now we want to know how much oil there is at the red point, without test drilling.

We call the geospatial process $Y(x)$ where x is a location set. In our example $y(x_i), i = 1, \dots, 8$ are the realized values from all of the black points. We will refer to $y(x^*)$ as the unknown true value of the red point and $\hat{y}(x^*)$ as our prediction for the red point.

We predict using the following formula

$$\hat{y}(x^*) = \sum_{i=1}^n w_i y(x_i). \quad (1)$$

We take every known value, multiply it by some weight w_i , and then sum the result. Formula (1) is an example of a “linear” predictor, since the result is a linear combination of the data. At this point the objective is to find the weights, $\{w_i\}$. This will be discussed thoroughly in Section 2.4.

Figure 10: Kriging Example - Weights

A key constraint for (1) is that [4]

$$\sum_{i=1}^n w_i = 1.$$

Assuming that the mean of the process $Y(x)$ is a constant, μ , for all loca-

tions, the above formula will give us an unbiased predictor $\hat{y}(x^*)$, because

$$E[\hat{y}(x^*)] = E\left[\sum_{i=1}^n w_i y(x_i)\right] = \sum_{i=1}^n w_i E[y(x_i)] = \mu \sum_{i=1}^n w_i = \mu.$$

2.2 Stationarity

In order for Kriging to be effective the geospatial process $Y(x)$ must be stationary. If a geospatial process is stationary then the correlation of two points is dependent only on the distance and not on absolute location [4]. Suppose we have two points x_1 and x_2 . If $Y(x)$ is stationary then $E[Y(x_1)] = E[Y(x_2)]$ and $Cov(Y(x_1), Y(x_2)) = Cov(h = d(x_1, x_2))$ where h is the distance between x_1 and x_2 . We say $Cov(h)$ since the distance is the only factor which affects the covariance if the process is stationary.

2.3 Variogram

2.3.1 Definition

Kriging uses information from the correlation structure at different lags. A variogram is simply a function of covariance. A variogram, for some lag h , given a stationary process is defined as γ where

$$\begin{aligned} \gamma(h) &= \frac{1}{2} E[(Y(x) - Y(x+h))^2] \\ &= \frac{1}{2} [Var(Y(x) - Y(x+h))] \\ &= \frac{1}{2} [Var(Y(x)) + Var(Y(x+h)) - 2Cov(Y(x), Y(x+h))] \\ &= \frac{1}{2} Cov(Y(x), Y(x)) + \frac{1}{2} Cov(Y(x+h), Y(x+h)) - Cov(Y(x), Y(x+h)) \\ &= \frac{1}{2} Cov(0) + \frac{1}{2} Cov(0) - Cov(h) \\ &= Cov(0) - Cov(h). \end{aligned} \tag{2}$$

It is important to keep in mind the relationship between the variogram and the covariance. One can see that as the covariance increases the variogram decreases since $Cov(0) \geq Cov(h)$. Therefore, larger variogram values imply less correlation.

2.3.2 Construction

To construct a variogram we first find the “semivariance”. The semivariance is defined as

$$\hat{\gamma}(h) = \frac{1}{2K} \sum_{h=d(x_i, x_j)} (y(x_i) - y(x_j))^2, \quad (3)$$

where K is the number of points satisfying $h = d(x_i, x_j)$.

In practice, a sample variogram is constructed by first creating a variogram cloud (Figure 11). The horizontal axis is the lag (distance) between two points. The vertical axis is the value resulting from (3), which is the squared difference between the response variables of the points. This is done for each pair of points (x_i, x_j) which results in a plot like that of Figure 11.

Figure 11: Variogram Cloud

To create a sample variogram we first choose fixed intervals along the horizontal axis, these are called “buckets”. We then calculate the average semivariance value within each bucket. Below in Figure 12 we see buckets with a distance range of 25 are used. The sample variogram gives information on the correlation structure of the process. To use such information in Kriging we typically fit a smooth curve to the sample variogram, this is called the “variogram model”. We then incorporate this variogram model in Kriging to produce predictions.

Figure 12: Sample Variogram and Variogram Model

The fitted line in Figure 12 is a variogram. To be a valid variogram it must be a conditionally negative semi-definite (CNSD) function since the covariance function is positive semi-definite (PSD) [11]. That is $\gamma(h)$ satisfies $\sum_{i=1}^n \sum_{j=1}^n \gamma(x_i, x_j) v_i v_j \leq 0$ where $\sum_{i=1}^n v_i = 0$. We provide the proof for this on the next page.

Proof:

We are given that

$$\gamma(x_i, x_j) = \frac{1}{2}[Cov(Y(x_i), Y(x_i)) + Cov(Y(x_j), Y(x_j)) - 2Cov(Y(x_i), Y(x_j))]$$

$$\begin{aligned} & \sum_{i=1}^n \sum_{j=1}^n v_i \gamma(x_i, x_j) v_j \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n [Cov(Y(x_i), Y(x_i)) + Cov(Y(x_j), Y(x_j)) - 2Cov(Y(x_i), Y(x_j))] v_i v_j \\ &= \frac{1}{2} \sum_{i=1}^n Cov(Y(x_i), Y(x_i)) v_i \sum_{j=1}^n v_j + \frac{1}{2} \sum_{i=1}^n v_i \sum_{j=1}^n Cov(Y(x_j), Y(x_j)) v_j - \\ & \quad \sum_{i=1}^n \sum_{j=1}^n Cov(Y(x_i), Y(x_j)) v_i v_j \\ &= - \sum_{i=1}^n \sum_{j=1}^n Cov(Y(x_i), Y(x_j)) v_i v_j, \text{ since } \sum_{i=1}^n v_i = 0 \end{aligned}$$

It is well known that $Cov(Y(x_i), Y(x_j))$ is positive semi-definite, so

$$- \sum_{i=1}^n \sum_{j=1}^n Cov(Y(x_i), Y(x_j)) v_i v_j \leq 0$$

□

Thus the choice of function for the variogram must be CNSD since the covariance function is PSD.

2.3.3 Models and Properties

Choosing a variogram model to fit the sample variogram (the points constructed from the variogram cloud) can be a challenging task. There are many different types of variogram models. We introduce two such models that have been used in our analysis.

Exponential Model: $\gamma(h) = k\mathbf{1}(h)_{(0,\infty)} + c[1 - \exp(-|h|/r)]$

The k in the above equation is called a nugget; it shifts the variogram up vertically. c is the sill; it sets an upper bound for the variogram. In the exponential model we see that as the lag (h) goes off to infinity the variogram approaches $k + c$, the nugget plus the sill. The r is the scaling parameter; it will change the curvature of the model. Figure 12 is an example of an exponential variogram with a nugget of 50, a sill of 250 and a scaling parameter of 0.5.

The reason for the indicator beside k is to ensure that $\gamma(0) = 0$. This property must hold true. Recall $\gamma(h) = Cov(0) - Cov(h)$. Therefore, $\gamma(0) = Cov(0) - Cov(0) = 0$.

Periodic Model: $\gamma(h) = k\mathbf{1}(h)_{(0,\infty)} + c[1 - \cos(2\pi|h|/\omega)]$

This model generates a periodic variogram. The only new parameter compared to the exponential model is the ω which sets the periodicity of the model. We use this model frequently as we observed periodic sample variograms in our data.

2.4 Finding the Weights in Kriging

We now continue from the Kriging formula (1) in Section 2.1. Recall that we set up our prediction for $y(x^*)$ as

$$\hat{y}(x^*) = \sum_{i=1}^n w_i y(x_i),$$

where x^* is a new location. The goal of Kriging is to find the weights $\{w_i\}$ such that $\hat{y}(x^*)$ is an unbiased predictor with minimum squared prediction error

$$Err(w_i) = E[(\hat{y}(x^*) - y(x^*))^2].$$

Recall that unbiasedness is ensured by $\sum_{i=1}^n w_i = 1$ so to achieve the this we take,

$$\hat{w}_i = \underset{w_i}{\operatorname{argmin}} Err(w_i) \quad \text{subject to} \quad \sum_{i=1}^n w_i = 1.$$

To solve the above minimization problem, we represent $Err(w_i)$ in terms of the variogram.

$$\begin{aligned} & \left(\sum_{i=1}^n w_i y(x_i) - y(x^*) \right)^2 \\ &= \sum_{i=1}^n w_i y(x_i) \sum_{j=1}^n w_j y(x_j) - 2 \sum_{i=1}^n w_i y(x_i) y(x^*) + y(x^*)^2 \\ &= \sum_{i=1}^n \sum_{j=1}^n w_i w_j y(x_i) y(x_j) - \sum_{i=1}^n w_i y(x_i)^2 + \sum_{i=1}^n w_i (y(x_i) - y(x^*))^2 \end{aligned}$$

$$\begin{aligned}
&= \sum_{i=1}^n \sum_{j=1}^n w_i w_j y(x_i) y(x_j) - \sum_{i=1}^n \sum_{j=1}^n w_i w_j y(x_i)^2 + \sum_{i=1}^n w_i (y(x_i) - y(x^*))^2 \\
&= -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_j w_i (y(x_i) - y(x_j))^2 + \sum_{i=1}^n w_i (y(x_i) - y(x^*))^2
\end{aligned}$$

Taking expectation on both sides and recalling the definition of the variogram from (2), we have

$$Err(w_i) = -\sum_{i=1}^n \sum_{j=1}^n w_j w_i \gamma(x_i, x_j) + 2 \sum_{i=1}^n w_i \gamma(x_i, x^*) = -\mathbf{w}' \Gamma \mathbf{w} + 2\mathbf{w}' \boldsymbol{\gamma}^*. \quad (4)$$

Note that $Err(w_i)$ depends on the variogram $\gamma(x_i, x_j)$. In practice we will use the variogram model obtained from the sample variogram as described in Sections 2.3.2 and 2.3.3. We then minimize (4) subject to the constraint $\sum_{i=1}^n w_i = 1$. This is done using the method of Lagrange multipliers using the following auxiliary function [4]

$$\Lambda(w_i, \lambda) = Err(w_i) + \lambda \left(\sum_{i=1}^n w_i - 1 \right),$$

where λ is the Lagrange multiplier. To find the minimum point, we take the derivative of $\Lambda(w_i, \lambda)$ with respect to w_i and λ , equate them to 0, and then solve the equations for $\{w_i\}$ and λ . These equations are

$$\begin{aligned}
\frac{\partial \Lambda(w_i, \lambda)}{\partial w_i} &= -2\Gamma \mathbf{w} + 2\boldsymbol{\gamma}^* + \lambda = 0, \\
\frac{\partial \Lambda(w_i, \lambda)}{\partial \lambda} &= \mathbf{w}' \mathbf{1}_{n \times 1} = 1.
\end{aligned}$$

Writing them in matrix form, we get

$$\begin{bmatrix} 2\gamma(x_1, x_1) & 2\gamma(x_1, x_2) & \cdot & \cdot & \cdot & 2\gamma(x_n, x_1) & 1 \\ 2\gamma(x_2, x_1) & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 2\gamma(x_n, x_1) & \cdot & \cdot & \cdot & \cdot & 2\gamma(x_n, x_n) & 1 \\ 1 & 1 & \cdot & \cdot & \cdot & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_n \\ -\lambda \end{bmatrix} = \begin{bmatrix} 2\gamma(x_1, x^*) \\ 2\gamma(x_2, x^*) \\ \cdot \\ \cdot \\ \cdot \\ 2\gamma(x_n, x^*) \\ 1 \end{bmatrix}.$$

The final weights $\{\hat{w}_i\}$ are obtained by solving the system of linear equations. The final prediction then is given by

$$\hat{y}(x^*) = \sum_{i=1}^n \hat{w}_i y(x_i).$$

The set of weights $\{\hat{w}_i\}$ are based on the variogram; they do not depend on the values $y(x_i)$. Therefore, $\hat{y}(x^*)$ is a valid linearly unbiased predictor of $P(y(x^*) = 1)$.

2.5 Universal Kriging

The above formulation for Kriging is based on the assumption that the response process $Y(x)$ is stationary. When $Y(x)$ does not have a constant mean, we can use Universal Kriging to obtain an unbiased predictor. The idea is to model the non-stationary mean field with a linear regression based on some covariates. In general, we assume that

$$E[Y(x)] = \sum_{k=1}^P \beta_k f_k(x),$$

where f_k is the value of the k^{th} covariate at location x .

We still look for the best linear unbiased predictor of the form $\hat{y}(x^*) = \sum_{i=1}^n w_i y(x_i)$. However in this case, to ensure unbiasedness we cannot use $\sum_{i=1}^n w_i = 1$ anymore since the mean field is not stationary. Instead, we have to obtain an alternate constraint on $\{w_i\}$ to ensure unbiasedness [8]. Note that,

$$\begin{aligned}
E[\hat{y}(x^*)] &= E\left[\sum_{i=1}^n w_i y(x_i)\right] \\
&= \sum_{i=1}^n w_i E[y(x_i)] \\
&= \sum_{i=1}^n w_i \sum_{k=1}^P \beta_k f_k(x_i) \\
&= \sum_{k=1}^P \beta_k \sum_{i=1}^n w_i f_k(x_i).
\end{aligned}$$

Combining the above result and the assumed model $E[\hat{y}(x^*)] = \sum_{k=1}^P \beta_k f_k(x^*)$ we have that the constraint

$$\sum_{i=1}^n w_i f_k(x_i) = f_k(x^*), \quad k = 1, \dots, P \quad (5)$$

ensures the unbiasedness of $\hat{y}(x^*)$.

Since the predictor is still a linear combination of $y(x_i)$, the prediction error is the same as it was in in Section 2.4, i.e.

$$Err_{uk}(w_i) = -\sum_{i=1}^n \sum_{j=1}^n w_j w_i \gamma(x_i, x_j) + 2 \sum_{i=1}^n w_i \gamma(x_i, x^*) = -\mathbf{w}' \Gamma \mathbf{w} + 2\mathbf{w}' \boldsymbol{\gamma}^*.$$

To minimize $Err_{uk}(w_i)$ under constraint (5), we introduce the Lagrangian function,

$$\Lambda(w_i, \lambda_k) = Err_{uk}(w_i) + \sum_{k=1}^P \lambda_k \left(\sum_{i=1}^n w_i f_k(x_i) - f_k(x^*) \right).$$

Again, to find the minimum point, we solve the following equations [8]

$$\begin{aligned} \frac{\partial \Lambda(w_i, \lambda_k)}{\partial w_i} &= -2\Gamma \mathbf{w} + 2\boldsymbol{\gamma}^* + \mathbf{f}(\mathbf{x})\boldsymbol{\lambda} = 0, \\ \frac{\partial \Lambda(w_i, \lambda_k)}{\partial \lambda_k} &= \mathbf{w}' \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}^*). \end{aligned}$$

In practice, we solve the following system of linear equations for the final weights $\{\hat{w}_i\}$,

$$\begin{bmatrix} 2\gamma(x_1, x_1) & 2\gamma(x_1, x_2) & \cdot & \cdot & \cdot & 2\gamma(x_n, x_1) & f_1(x_1) & f_2(x_1) & \cdot & \cdot & \cdot & f_P(x_1) \\ 2\gamma(x_2, x_1) & \cdot & \cdot & \cdot & \cdot & \cdot & f_1(x_2) & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 2\gamma(x_n, x_1) & \cdot & \cdot & \cdot & \cdot & 2\gamma(x_n, x_n) & f_1(x_n) & \cdot & \cdot & \cdot & \cdot & f_P(x_n) \\ f_1(x_1) & f_1(x_2) & \cdot & \cdot & \cdot & f_1(x_n) & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ f_2(x_1) & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ f_P(x_1) & \cdot & \cdot & \cdot & \cdot & f_P(x_n) & 0 & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_n \\ -\lambda_1 \\ -\lambda_2 \\ \cdot \\ \cdot \\ \cdot \\ -\lambda_P \end{bmatrix} = \begin{bmatrix} 2\gamma(x_1, x^*) \\ 2\gamma(x_2, x^*) \\ \cdot \\ \cdot \\ \cdot \\ 2\gamma(x_n, x^*) \\ f_1(x^*) \\ f_2(x^*) \\ \cdot \\ \cdot \\ \cdot \\ f_P(x^*) \end{bmatrix}.$$

The resulting Universal Kriging predictor of $y(x^*)$ is given by

$$\hat{y}(x^*) = \sum_{i=1}^n \hat{w}_i y(x_i).$$

3 Network Kriging

We take the concept of Kriging, which is performed in a Euclidean space, and implement it in a network. We already stated that the weights of the links in our network are binary so distances can only take integer values. As consequence the corresponding variogram is defined for integer lags only. This is a fundamental difference between Network Kriging and ordinary Kriging. As well, the response variable (the role of the node of interest) $Y(x)$ is also binary. The response variables in Kriging are typically a continuous variable. In our case $Y(x)$ is a discrete process.

3.1 Network Kriging Method

First, we define a distance between any two nodes in the network. For the purpose of our method we will say that the distance between any two nodes is the distance of the shortest path between those two nodes. This means our variogram is defined for lags with discrete values ranging from 1 up to and including the diameter of the network.

Another feature of our data that will not permit a straightforward application of Kriging is that the variable to be predicted is a binary variable instead of a continuous variable. If we still use prediction of the form $\hat{y}(x^*) = \sum_{i=1}^n w_i y(x_i)$ without further constraints on $\{w_i\}$, $\hat{y}(x^*)$ may return a value outside the range of $[0, 1]$. Such prediction does not allow for interpretation of results. To avoid this we add the following constraint to Kriging,

$$0 \leq \sum_{i=1}^n w_i y(x_i) \leq 1.$$

Adding this constraint the final prediction of $\hat{y}(x^*)$ is a value within $[0, 1]$ and is interpreted as the predicted probability that $\hat{y}(x^*)$ takes value 1.

As a summary, to find the prediction weights $\{w_i\}$ in Network Kriging, we solve the following minimization problem.

$$\begin{aligned} & \underset{w_i}{\operatorname{argmin}} E[(\sum_{i=1}^n w_i y(x_i) - y(x^*))^2] \\ & \text{subject to } \sum_{i=1}^n w_i = 1, \quad 0 \leq \sum_{i=1}^n w_i y(x_i) \leq 1. \end{aligned}$$

This is a quadratic programming problem. We are trying to find the minimum of a convex function subject to a boundary created by the constraints. The *ConstOptim* function in R is able to handle this type of problem. It uses an adaptive barrier algorithm to find the minimum.

In many instances a variogram was constructed and used for prediction successfully. Unfortunately, fitting variograms can be laborious work. To speed up this process we decided in some circumstances to use the sample variogram points as our variogram values. Since we have a discrete space we will always have information at each lag value; unless there are only two nodes which create the diameter of the network and one of those two nodes is the node of interest. We did assessments using both the sample variogram and a fitted variogram and found the results to be extremely similar. Therefore in some instances for computational efficiency the sample variogram was used.

3.2 Network Stationarity

As we have discussed, the effectiveness of Kriging is dependent on stationarity. We address the stationarity issues in Network Kriging using three different methods.

3.2.1 Universal Network Kriging

Recall that Universal Kriging is applied when there is a covariate which is highly influential on the predictor variable (the mean of the process is not constant at all locations). In our case, we will use Universal Network Kriging when we believe there is a covariate which is highly correlated with the role of interest.

As before Universal Network Kriging is the same as Universal Kriging with the addition of our constraint which restricts results to fall within $[0, 1]$. Therefore we have the following optimization problem

$$\begin{aligned} & \underset{w_i}{\operatorname{argmin}} E[(\sum_{i=1}^n w_i y(x_i) - y(x^*))^2] \\ & \text{subject to } \sum_{i=1}^n w_i f(x_i) = f(x^*), \quad 0 \leq \sum_{i=1}^n w_i y(x_i) \leq 1. \end{aligned}$$

We have a quadratic programming problem as we did with Network Kriging. We can once again solve this using the *ConstOptim* function in R. Here we only use $f(x_i)$ instead of $f_k(x_i)$ as we will only be using one covariate.

3.2.2 Cluster Network Kriging

One way we will try to remedy stationarity issues is by performing Network Kriging in subnetworks. That is, we break the network into stationary clusters and then perform Cluster Network Kriging (CNK). This method will be effective if the correlation structure of the role is dependent on the cluster in which the node falls.

To parse the network we find the clusterings which maximize the modularity of the network [12]. Modularity is a metric proposed by Newman to measure how well a network is partitioned into clusters. It allows one

to evaluate different clusterings of the same network by comparing their modularities.

The modularity of an undirected network will be defined by Q [12] where $Q = (\text{fraction of links within clusters}) - (\text{expected fraction of links within clusters})$.

3.2.2.1 Modularity - Two Clusters

We start with the simplest case, modularity for an undirected network with two clusters. To do this we need to find the fraction of links within clusters and the expected fraction of links within clusters.

$$\sum_{i=1}^n \sum_{j=1}^n A_{i,j} = 2m$$

will return two times the number of links in the network (m). This is because the network is undirected, resulting in each link being counted twice. Therefore,

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n A_{i,j} = m$$

will now return the number of links in the network. Now consider,

$$\frac{1}{2m} \sum_{i=1}^n \sum_{j=1}^n A_{i,j} \frac{(s_i s_j + 1)}{2}. \quad (6)$$

By dividing by $2m$ in (6) we are dividing by the sum of the entire adjacency matrix. $\frac{(s_i s_j + 1)}{2}$ will return a 1 or a 0 as s_i is 1 if node x_i is in cluster one or -1 if in cluster two. Thus if both x_i and x_j are in the same cluster: $\frac{((1)(1)+1)}{2} = 1$ or $\frac{((-1)(-1)+1)}{2} = 1$ versus if x_i and x_j are in different clusters: $\frac{((-1)(1)+1)}{2} = 0$ or $\frac{((1)(-1)+1)}{2} = 0$. The result is that equation (6) now only adds the links which are in the same cluster and subsequently this returns the fraction of links within clusters.

Now that we have the fraction of links within clusters, we must calculate the expected fraction of links within clusters. Recall the degree centrality of a node is the number of links which are connected to the node, i.e.

$$D(x_i) = \sum_{j=1}^n A_{i,j}.$$

For a randomly distributed network,

$$\frac{D(x_i)D(x_j)}{2m}$$

returns the expected number of links between nodes x_i and x_j . To get the expected fraction of links within a cluster we do the following,

$$\frac{1}{2m} \sum_{i=1}^n \sum_{j=1}^n \frac{D(x_i)D(x_j)}{2m} \frac{(s_i s_j + 1)}{2}. \quad (7)$$

Once again, we divide by $2m$ to get the fraction of links rather than the number of links. As well, $\frac{(s_i s_j + 1)}{2}$ only calculates the measure when the nodes are in the same cluster. Now we subtract equation (7) from equation (6) to get the modularity [12]

$$Q = \frac{1}{2m} \sum_{i=1}^n \sum_{j=1}^n A_{i,j} \frac{(s_i s_j + 1)}{2} - \frac{1}{2m} \sum_{i=1}^n \sum_{j=1}^n \frac{D(x_i)D(x_j)}{2m} \frac{(s_i s_j + 1)}{2}.$$

This can be simplified to

$$Q = \frac{1}{4m} \sum_{i=1}^n \sum_{j=1}^n (A_{i,j} - \frac{D(x_i)D(x_j)}{2m})(s_i s_j + 1), \quad (8)$$

and even further to

$$Q = \frac{1}{4m} \sum_{i=1}^n \sum_{j=1}^n (A_{i,j} - \frac{D(x_i)D(x_j)}{2m})(s_i s_j). \quad (9)$$

The removal of the 1 at the end of (8) is easy to see after expanding

$$Q = \frac{1}{4m} \left[\sum_{i=1}^n \sum_{j=1}^n \left(A_{i,j} - \frac{D(x_i)D(x_j)}{2m} \right) (s_i s_j) + \sum_{i=1}^n \sum_{j=1}^n \left(A_{i,j} - \frac{D(x_i)D(x_j)}{2m} \right) \right].$$

Manipulating the second half of the equation,

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^n \left(A_{i,j} - \frac{D(x_i)D(x_j)}{2m} \right) &= \sum_{i=1}^n \sum_{j=1}^n A_{i,j} - \frac{1}{2m} \sum_{i=1}^n \sum_{j=1}^n D(x_i)D(x_j) \\ &= 2m - \frac{1}{2m} \sum_{i=1}^n D(x_i) \sum_{j=1}^n D(x_j) \\ &= 2m - \frac{1}{2m} 2m \cdot 2m \\ &= 0 \end{aligned}$$

Since the equation above works out to 0 all we are left with is equation (9). When calculating the modularity, equation (8) will be used as many of the terms will go to 0. Equation (9) will be useful for methods which maximize modularity [12].

Since the network is undirected, equation (8) is still doing more work than necessary. Because $A_{i,j} = A_{j,i}$ it is not necessary to go through all the possible pairs of nodes in the network but just the unique pairs and multiply by 2 when required, giving the following result

$$Q = \frac{1}{2m} \left[\sum_{i=1}^n \left(A_{i,i} - \frac{D(x_i)D(x_i)}{2m} \right) + \sum_{i=1}^n \sum_{j=1, i < j}^n \left(A_{i,j} - \frac{D(x_i)D(x_j)}{2m} \right) (s_i s_j + 1) \right]. \quad (10)$$

3.2.2.2 Two Cluster Modularity Maximization

To maximize modularity we return to the derivation given in equation (9) and rewrite it in matrix notation [12]

$$Q = \frac{1}{4m} \mathbf{s}' \mathbf{B} \mathbf{s}.$$

$\mathbf{B}$ is referred to as the modularity matrix [12]. Q must be maximized with respect to $\mathbf{s}$. In other words, we maximize Q by changing the cluster assignment. The values in $\mathbf{s}$ must be either ± 1 . This constraint makes the maximization problem difficult. The constraint is therefore altered so that $\sum_{i=1}^n s_i^2 = \mathbf{s}' \mathbf{s} = n$.

The auxiliary function is set up to maximize [12]

$$\Lambda(s_i, \lambda) = \frac{1}{4m} \mathbf{s}' \mathbf{B} \mathbf{s} + \lambda(\mathbf{s}' \mathbf{s} - n).$$

Taking partial derivatives we solve

$$\begin{aligned} \frac{\partial \Lambda(s_i, \lambda)}{\partial s_i} &= \frac{1}{2m} \mathbf{B} \mathbf{s} + 2\lambda \mathbf{s} = 0, \\ \frac{\partial \Lambda(s_i, \lambda)}{\partial \lambda} &= \mathbf{s}' \mathbf{s} - n = 0. \end{aligned}$$

We rearrange the partials with respect to $\mathbf{s}$ and ignore the constants as they do not affect the maximization to obtain

$$\mathbf{B} \mathbf{s} = \lambda \mathbf{s}. \tag{11}$$

There should be a negative in front of the lambda, but since it is an arbitrary value its inclusion is not necessary. From (11) we see that $\mathbf{s}$ is an eigenvector and λ is an eigenvalue of $\mathbf{B}$. In order to maximize the modularity, the eigenvector corresponding to the largest eigenvalue is used.

The eigenvector cannot be directly used as its values will not meet the original constraint that the entries of $\mathbf{s}$ be ± 1 . Instead, if a value of the eigenvector which maximizes the modularity, called $\mathbf{u}$, is greater than or equal to 0, we assign it to 1 and if it is less than 0 it is assigned to -1. So $u_i \geq 0$ then $s_i = 1$ and $u_i < 0$ then $s_i = -1$

3.2.2.3 N Cluster Modularity Maximization

The modularity maximization for N clusters is just an extension of the method for two clusters. This is done by bisecting the network within clusters. Once the bisection has been found which maximizes the modularity, we continue bisecting within the new clusters. The process stops in a cluster when the modularity is maximized by placing all of the nodes within the same cluster. The formula for measuring the change in modularity for an entire network after a bisection is given by [12]

$$\Delta Q = \frac{1}{4m} \sum_{i \in c} \sum_{j \in c} (A_{i,j} - \frac{D(x_i)D(x_j)}{2m})(s_i s_j + 1) - \frac{1}{2m} \sum_{i \in c} \sum_{j \in c} (A_{i,j} - \frac{D(x_i)D(x_j)}{2m}), \quad (12)$$

c is the cluster number and n_c is the number of nodes within cluster c . The left half of (12) is simply equation (9) within cluster c . The right half is the present state of cluster c ; in other words it is the modularity if the cluster is not partitioned. Subsequently if an $\mathbf{s}$ can be found such that ΔQ is positive the cluster will be partitioned. If not it will stay the same and the algorithm will halt. The method for maximizing ΔQ follows exactly as it did in the two community case.

Using modularity to parse the network in our example resulted in the clusters in Figure 13.

Figure 13: Network after clustering

To predict the role of a node using Cluster Network Kriging we first identify the cluster, C , which contains the node of interest. Then we proceed with Network Kriging as usual, but using only the nodes which are members of the identified cluster. This results in the following altered semivariance,

$$\hat{\gamma}(h) = \frac{1}{2K_C} \sum_{h=d(x_i, x_j)} (y(x_i) - y(x_j))^2 \delta(x_i \in C) \delta(x_j \in C).$$

Kronecker's delta restricts the nodes to fall within the cluster of interest. The K_C is the number of points satisfying $h = d(x_i, x_j)$ in the cluster.

3.2.3 Neighbourhood Network Kriging

The final way we will try to remedy stationarity issues is by performing Neighbourhood Network Kriging (NNK). The logic is similar to CNK, however in this case the correlation structure of the role will be more dependent on the location of the node rather than the cluster in which it falls. This method will perform Network Kriging as usual but within a fixed neighbourhood of the node of interest. By doing this the semivariance changes from (3) to

$$\hat{\gamma}(h) = \frac{1}{2K_N} \sum_{h=d(x_i, x_j)} (y(x_i) - y(x_j))^2 \delta(x_i \in N) \delta(x_j \in N).$$

Kronecker's delta restricts the nodes to fall within the neighbourhood. The K_N is the number of points satisfying $h = d(x_i, x_j)$ in the neighbourhood.

Using CNK or NNK will create some issues of their own. These issues will be discussed in Section 3.2.4.

3.2.4 Remarks

Two issues can arise from performing either Cluster Network Kriging or Neighbourhood Network Kriging.

The first issue can be clearly seen in Figure 13. Some of the clusters contain a very small number of nodes. This means we will not have much data to construct a variogram. As well, the diameter of some of the clusters is small. In some instances a cluster had a diameter of three. Fitting a variogram to a sample variogram of size three is rather difficult. This will also be an issue if the neighbourhoods in NNK are small.

The second issue also has to do with a lack of information. Suppose the diameter of one of the clusters/neighbourhoods is three. However, the

only node which has a shortest path to other nodes of size three is x^* , the node which we are trying to predict. Subsequently, the sample variogram will not have any information at lag three. The variogram chosen will have theoretical values at lag three but properly choosing a variogram using only two sample points (at lags 1 and 2) is impossible.

To solve the first issue one could try different clustering methods to try and get larger clusters while still maintaining stationarity. Or, we could only use CNK on sufficiently large networks so that the clusters have enough nodes. With regards to NNK one would just have to make sure the neighbourhood is not too small.

If the first issue is resolved then the second issue would become moot, as the diameters of the clusters/neighbourhood would be large enough to see variogram structure. In our case, if we were trying to infer from a lag where no information was available we would not use that node for inference.

Another difficulty with this process is that a different variogram must be constructed for each cluster/neighbourhood. This process can become very time consuming.

3.3 Results

We provide the prediction results for four different roles in the network, namely Emir (Leadership), Finance/Logistics, Subordinate, and Fatwa Committee (Religious Leaders). In each instance we compare Network Kriging (or some version of it) to KNN and in some instances to logistic regression.

Methods will be evaluated using receiver operator characteristic (ROC) curves. The x-axis of the plot for an ROC curve is the false positive (FP) rate and the y-axis is the true positive (TP) rate. Therefore better curves are more to the upper-left hand corner; meaning that they have a higher TP rate for a lower FP rate [5]. ROC measures the TP and FP rate of a binary prediction at all possible thresholds. So it is usually regarded as the most complete performance measure for the prediction of a binary variable. The ROC curve is created by plotting the TP rate against the FP rate for a large number of thresholds.

3.3.1 Emir (Leadership)

Figure 14: 1998 - Emir

The red nodes in Figure 14 are the Emirs in the network. There are four Emirs in total, three of which are clustered at the top of the network. We would not expect Network Kriging to perform very well in this instance. There are too few Emirs in the network to gain significant variogram structure. We checked to see if any of our centrality metrics have a correlation

with Emir. If one of the metrics is highly influential of the role then Universal Network Kriging would be a good option. As it turned out, degree centrality (DC) was a strong candidate, so we use it as a covariate.

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-0.07805	0.01450	-5.383	2.63e-07	***
DC	0.02041	0.00210	9.720	< 2e-16	***

Overall the semivariance values seen in Figure 15 are fairly low regardless of the lag. In our circumstance this means that there is either an abundance or a small number of nodes with the role of interest. There are obviously only a small number of Emirs in the network. The variogram structure for Emir (Figure 15) shows a lower correlation at low lags and a higher correlation at larger lags. This would suggest that Emirs are more central in the network, which we can clearly see from Figure 14.

Figure 15: Variogram - Emir

Figure 16: ROC 1 - Emir

The ROC curves for the predictions using Universal Network Kriging and KNN are shown in Figure 16. Universal Network Kriging performed better than KNN. However, this result raises an interesting question. If the covariate in Universal Network Kriging is the reason why the prediction is working well then could we not drop Network Kriging and just perform a logistic regression with degree centrality as the predictor?

Figure 17: ROC 2 - Emir

The ROC curves for Universal Network Kriging and logistic regression are shown in Figure 17. As one can see, logistic regression is only able to find three out of four Emirs in the network. Universal Network Kriging is eventually able to find the fourth. This is because only three out of four leaders have a very high number of links while the fourth has only a moderate amount. Universal Network Kriging is able to find the fourth Emir using the positioning of the other Emirs in the network.

3.3.2 Finance/Logistics

We see from Figure 18 that the network structure for Finance/Logistics is very different from Leadership. There are a total of forty-one nodes in the network which fulfil the role of Finance/Logistics.

Figure 18: 1998 - Finance/Logistics

The variogram in Figure 19 shows a very interesting pattern. Typically variograms look like the one in Figure 12; they gradually increase towards an

upper bound. Kriging is used in geographic regions and as one would expect two regions that are farther apart are less correlated than two regions that are close. In the network we are not seeing that commonplace relationship. The variogram in our case results in a quadratic-like shape. This indicates that there are similarities with this role at low and high lags, while most variability occurs in the middle lags (from about lag 3 to 8).

Figure 19: Variogram - Finance/Logistics

This structure makes fitting variograms rather challenging. The vari-

ogram fitted in Figure 19 is from a periodic model. Around half a period was used to create the quadratic shape. Network Kriging and KNN were both used to predict whether or not a member fulfils the Finance/Logistics role. Figure 20 shows the ROC curves which resulted from the two methods.

Figure 20: ROC 1 - Finance/Logistics

Network Kriging seems to be performing rather poorly compared to KNN. This is most likely a result of a lack of stationarity. For a fixed lag (say lag 3), the correlation structure will be different depending on what area of the network is examined. When we construct a variogram we amalgamate

the correlation structure throughout the entire network; so if the network is not stationary the prediction performance will be poor. We attempt to remedy this issue by using Cluster Network Kriging and Neighbourhood Network Kriging. Universal Network Kriging is not a viable option in this case as there is no covariate which is correlated with the role.

Figure 21: ROC 2 - Finance/Logistics

The ROC curves for the predictions using NNK and CNK have been added and are shown in Figure 21. By doing Network Kriging locally (either cluster or neighbourhood based), we see improvements. However the performance is still lower than KNN.

3.3.3 Subordinate

There are a total of 63 nodes which fulfil to role of Subordinate.

Figure 22: 1998 - Subordinate

In Figure 23 we see that, similar to Finance/Logistics, there tends to be the most variation in the middle lags with Subordinate. This is evident from Figure 22 as we see tight clusters of Subordinates on opposite sides of the network.

Figure 23: Variogram - Subordinate

Once again we predict using both KNN and Network Kriging. The resulting ROC curves are plotted below in Figure 24.

Figure 24: ROC 1 - Subordinate

As seen by the ROC curves in Figure 24, Network Kriging does not seem to be performing as well as KNN. However, there does seem to be an improvement with the Subordinate role over Finance/Logistics. This is most likely a combination of there being more Subordinates in the network (which gives more information to construct a variogram) and the Subordinate role being more stationary. We try to improve the results by using Cluster Network Kriging and Neighbourhood Network Kriging.

Figure 25: ROC 2 - Subordinate

Figure 25 shows the ROC curves of both CNK and NNK for the Subordinate role. This time neither technique was able to improve the results of Network Kriging.

3.3.4 Fatwa Committee

There are only 6 nodes in the network which are members of the Fatwa Committee. As was the case with Emir, we find that degree centrality is correlated with the role. This means Universal Network Kriging should be a good option.

Figure 26: 1998 - Fatwa Committee

Figure 27: Variogram - Fatwa Committee

The variogram structure for the Fatwa Committee is very similar to Emir. There is stronger correlation at larger lags. This suggests like Emir, that Fatwa Committee members are more central in the network.

Figure 28: ROC - Fatwa Committee

Universal Network Kriging has clearly performed much better than KNN as seen by the ROC curves in Figure 28.

3.4 Remarks

Initial assessments of Network Kriging show promise. Network Kriging out performed both KNN and logistic regression for roles which were rare in the network such as Emir and Fatwa Committee. This is a very positive result as roles like these are very important to identify in a terrorist network. More specifically these roles were identified using Universal Network Kriging. Universal Network Kriging used both a covariate and the location of the node in the network to give strong prediction performance.

Network Kriging was not as effective as KNN for roles which were commonplace in the network such as Finance/Logistics and Subordinate. We believe the main reason behind the lesser performance was a lack of stationarity. That is, the correlation structure of the nodes is dependent on the area of the network. We attempted to remedy this issue by performing Network Kriging within a cluster and separately within a neighbourhood. These methods either improved the performance of Network Kriging or the performance was unchanged. However, neither method was able to perform quite as well as KNN and in some instances these methods created problems of their own, as discussed in Section 3.2.4. Some analysis was done to visualize the stationarity issue.

Figure 29: Subnetwork Variograms - Emir

In Figure 29, we see several sample variograms constructed from different regions of the network for the role of Emir. Overall, there is a fairly consistent trend in each subnetwork. This shows that this variable is fairly stationary. There are a few instances where the sample variograms are flat at 0. This simply indicates that all of the nodes in that area of the network either fulfil the role or all of the nodes do not fulfil the role. In the case of Emir, there are subsections of the network where there are no Emirs.

Figure 30: Subnetwork Variograms - Finance/Logistics

In Figure 30, we see more variability in the structure of the sample variograms for the role of Finance/Logistics than we did for Emir. This implies that the role of Finance/Logistics is less stationary than the role of Emir. Moreover, we would not expect Network Kriging to perform as well for the role of Finance/Logistics, which was the case in our results. It also explains why CNK and NNK improved the results of Finance/Logistics. By isolating the area of prediction to be local to the node of interest we use the unique variogram generated from that area of the network.

Figure 31: Subnetwork Variograms - Subordinate

With Subordinate we see (Figure 31) some variability in the sample variograms. There is more variability than Emir, but not as much as Finance/Logistics. This follows the trend as Subordinate produced results better than Finance/Logistics, but not as well as Emir. Using CNK and NNK with Subordinate resulted in no change in performance. We assumed to see an improvement as there does seem to be some variability in the sample variograms. However, the marginal improvement in performance was probably negated by some of the issues discussed in Section 3.2.4.

Figure 32: Subnetwork Variograms - Fatwa Committee

Like Emir, Fatwa Committee has fairly stable sample variograms. Subsequently, there seemed to be no major issues with the prediction performance.

4 Conclusion

The purpose of this thesis was to develop a method which was effective at predicting roles in a network. After some initial exploratory analysis we decided to modify the geostatistical prediction method of Kriging to suit our needs.

Implementing Kriging in a network presented challenges. On occasion, the correlation structure of the network was not stationary. We attempted to remedy this issue by clustering the network and then performing Network Kriging within each cluster, or a similar method with neighbourhoods. However, these method presented their own set of problems. There were instances where clusters had a very small number of nodes, giving us variograms based on little information. The methodology did result in an increase in prediction accuracy for some roles.

We addressed other stationarity issues by using Universal Network Kriging. This was done by finding a covariate which was highly influential of the role of interest. By using this covariate we stabilized the mean field, allowing Network Kriging to be effective.

This thesis shows that under certain circumstances Network Kriging is a viable option for prediction. In the future we would like to try Network Kriging on a larger network with a variety of variables. Kriging is by no means restricted to predicting a binary variable. It would be interesting to use Network Kriging on a continuous variable which has a geospatial undertone, such as a disease outbreak. We would also like to add a temporal component in order to predict in a dynamic network.

References

- [1] H. Abolhassani and M. Jamali. *Different Aspects of Social Network Analysis*. Sharif University. Accessed: July 3, 2013. Website: <https://www.cs.sfu.ca/~oschulte/teaching/socialnetwork/papers/SNA-intro-mohsen.pdf>
- [2] J. Cao, B. Xia and J. Yuan. *Arresting Strategy Based on Dynamic Criminal Networks Changing over Time*. Southeast University, King Abdulaziz University and New Star Institute of Applied Technology. Accessed: June 15, 2013. Website: <http://www.hindawi.com/journals/ddns/2013/296729/>
- [3] K. Carley. *Dynamic Social Network Modeling and Analysis*. Page: 133-145. Carnegie Mellon University. Accessed: June 28, 2013. Website: http://www.nap.edu/openbook.php?record_id=10735
- [4] N. Cressie, C. K. Wikle. *Statistics for Spatio-Temporal Data*. John Wiley & Sons, New Jersey, 2011.
- [5] T. Fawcett. *An Introduction to ROC Analysis*. Institute for the Study of Learning and Expertise. Website: <https://ccrma.stanford.edu/workshops/mir2009/references/ROCintro.pdf>. Accessed: January 15, 2016.
- [6] L. Getoor. *Link Mining: A New Data Mining Challenge*. University of Maryland. Accessed: July 13, 2013. Website: <http://citeseerx.ist.psu.edu/vi>
- [7] H. Heidemann, B. Friedl and A. Landherr. *A Critical Review of Centrality Measures in Social Networks*. Augsburg University. Accessed: July 24, 2013. Website: <http://www.wi-if.de/paperliste/paper/wi-282.pdf>

- [8] J. Janicze. *Universal Kriging in Multiparameter Transducer Calibration*. Wroclaw University of Technology. Website: http://www.metrology.pg.gda.pl/full/2009/M&MS_2009_661.pdf. Accessed: September 29, 2015.
- [9] E. Kolaczyk. *Statistical Analysis of Network Data - Methods and Models*. Springer, New York, 2009.
- [10] D. G. Krige A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa* vol 52, 119-139, 1951.
- [11] K. Loquin and D. Dubois. *Kriging and Epistemic Uncertainty: a Critical Discussion*. Université Paul Sabatier. Website: https://www.irit.fr/~Didier.Dubois/Papers1208/fuzzy_Kriging-livre1.pdf. Accessed: September 3, 2015.
- [12] M. E. J. Newman. *Networks An Introduction*. Oxford University Press, Oxford, 2010.
- [13] B. Srinivasan, R. Duraiswami and R. Murtugudde. *Efficient kriging for real-time spatio-temporal interpolation*. University of Maryland. Accessed: September 25, 2015. Website: <http://www.climateneeds.umd.edu/pdf/EfficientKrigingforReal-Time.pdf>