UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Geometrical Social Networks

Jayanth Chinthu Rukmani Kumar

M.Sc. Thesis Individual research assignment
July 2021

Supervisors:
Prof. Dr. Peter Lucas
Dr. Clara Stegehuis

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

The rise of social networks in the past few years has been remarkable. The main reason behind the tremendous success of the social media is none other than Facebook. Facebook paved the way for other popular social networking applications such as Twitter and Instagram. Facebook is an online social networking application where users can become friends and socialize with anyone throughout the world. To find how two users from various different regions are connected, Facebook came with an idea called the Social Connectedness Index (SCI) to quantitize the strength of the connection. We expected that the relationship between users and their friend, the SCI, and the distance between users and their friends has some characteristic dependence and independence information. Such independence information is nowadays often represented by means of Bayesian networks. A Bayesian network is a type of probabilistic graphical model that captures dependence and independence information as a graph and a joint probability distribution.

In this research, the main emphasis is on finding the right structure of the Bayesian network for predicting the SCI. In particular, we focus on structure learning and parameter learning from data. Tabu search is one of the structure learning algorithms we used to find right structure of the Bayesian network. The resulting network is subsequently validated using K-fold cross-validation, confusion matrices, ROC analysis, and calibration plots to see the difference between observed and predicted probabilities. In addition, the Brier score and the Log-likelihood of the model are calculated. We also compared the results of the model for various different group of countries and also investigated the distribution of SCI and distance between users and friends. The next thing was to find the optimal number of bins when discretizing the data. Finally, we compared the Tabu-search Bayesian network with a naive Bayes network.

We were able to find the right Bayesian network structure using Tabu search. When the network is validated using K-fold cross-validation, the scores we get from that show that the network has performed well. The accuracy obtained is 87 percentage which again shows that the model has performed well and the ROC curve is an other example for showing how well the model has performed. The Brier score obtained is close to 0 which again proves the worth of the model. When the model is compared to the Naive Bayes network, the Naive Bayes network achieved an accuracy of about 30 percentage and it has performed poorly when we compare it to the Tabu search based model. The calibration plots of scaledsci and distancekm show that the model has calibrated very well. When the model is compared for various different group of countries, the distribution of scaledsci and distancekm changes for every particular group of countries. The optimal number of bins chosen is 4 since for every other bin, the network encountered problems like low accuracy and data imbalancing.

Keywords: Social Connectedness Index, Bayesian networks, social networks

Table of Contents

1. Introduction and Related work

1.1 Motivation

Social networks have become so popular that they are involved in almost every person's life. Facebook, Instagram and Twitter are some of the most popular social networking sites. People use social networks to build social relationships with other people who might have similar interests, activities and careers. Social networks make people share their ideas, photos, videos and also inform other people about real-world activities with their friends.

Facebook is one of the major reasons for the success of online social media. It was founded in the year 2004, by Mark Zuckerberg. It is basically a website where every user can register and create a free account. After creating a Facebook profile, each user can introduce themselves by filling in required information, such as updating their profile picture and biography.They can also share their thoughts with others. They can also connect with others by sending a friend request. The person to whom they are sending a friend request can be from any part of the world. Facebook also provides a necessary messaging platform called Facebook Messenger, where every user can chat with their friends. They can also share a variety of information with their Facebook Messenger friends, such as photos, videos, stickers, audios, and files.

Many social networking sites struggle to understand their social connections.So, to overcome this and also to understand how people from different geographical regions are connected, the research group on Facebook, led by Michael Bailey and his fellow colleagues, came with an idea which is known as the 'Social Connectedness Index' (SCI). As a result, in this study, we will build a Bayesian network and see what kinds of effects and insights can be obtained from it.

1.2 Social connectiveness

The **Facebook Social Connectedness Index** (**SCI**) is used for measuring the strength of connectedness among any two regions. The SCI gives variety of information like economic opportunities, social mobility and trade. Using the SCI index, we can measure the connectivity between two people living in two different regions.

SCI is built using the information available from the friendship links between between all Facebook users as of April 2016. It was reported that 58 percentage of United States (US) adult population and 71 percentage of US online population use Facebook [1]. It is also reported that Facebook is common among the age group of 18-29 years old [1].

The Social Connectedness Index (SCI) is a new area of research which is used for computing the frequency and thickness of friendship and companions around the world. It is a type of data which is useful to find out how relationships accepts social outcomes. Since Facebook has more than 2.5 billion active users around the world, SCI delivers the first complete measure of social networks at a global level.

Probabilistic graphical model is a type of a probability model to represent the conditional dependences and independences between variables by means of a graph. They are most commonly used in statistical machine learning. It is possible to learn Bayesian networks from data, i.e. both network structure and probabilistic parameters can be learnt, and if that is done successfully the result offers a lot of insight into the problem domain. It is also possible to do probabilistic inference with a Bayesian network by computing the joint probability distribution of any subset of variables, conditioned on an instantiated other subset of variables.

So, in this thesis, the main intention is to find the right Bayesian network structure and check if it is suitable for this kind of problem. Furthermore, we also evaluate the network using a confusion matrix. We also validate the network using calibration curves, ROC and K-fold cross-validation. We also use log-likelihood to check the goodness of the fit of the model. We also compare the network with a Naive Bayes network. We also check how two variables are distributed. We also check the accuracy of the model using Brier score.

The type of Bayesian network which we are going to construct is going to be based on Structure learning. Learning a Bayesian network from the data has two major tasks: learning the structure of the network (structure learning) and learning the parameters (parameter learning). In this research, our will focus is on Structure learning. When we use data to learn the links of our Bayesian networks, then the type of learning is called Structure learning. We will construct a Tabu search-based Bayesian network.

1.3 Research questions

The uncertain relationship between two or more variables can be best represented by a probabilistic graphical model. Bayesian networks which belong to probabilistic graphical models have the advantage of having been investigated extensively. A straightforward Bayesian network is the so-called *naive* Bayes network. It has a fixed structure and makes strong conditional independence assumptions. Nowadays, both network structure and probabilistic parameters of a Bayesian network can also be learnt from data. It is unclear whether this can be done for Facebook data.

This brings us to the following research questions:

- Is it possible to find the right Bayesian network structure from Facebook data such that it can be used to predict the SCI?
- Assuming that we are able to learn a Bayesian network, yields some subsequent research questions:
  - How can such a model be validated?
  - What is the performance compared to a naive Bayes network?
  - Is it possible to obtain any useful information from the model by comparing the models for various groups of countries?
  - What are the optimal number of bins using discretization?

Thus, for answering the research question, we will try to find the right Bayesian network for predicting the SCI between two countries. In particular, we will try to learn a Bayesian network using Tabu search Bayesian network and see if it finds the right structure. We will also evaluate the network by means of a Confusion matrix and ROC curve. We also validate the network using K-Cross validation and by constructing calibration plots. We will also find the Brier score of the model to see if the model has performed well. We will also compare the model with the Naive Bayes model and evaluate both the Tabu based search and Naive Bayes by means of a confusion matrix and Log-Likelihood. We will also try to find out if any useful information can be extracted from the model by comparing the model for different groups of countries and this will also help us in seeing the probability distribution of scaledsci and distancekm for various group of countries. We will also try to find the optimal number of bins for intervals of discretization.

1.4 Related work

1.4.1 Social networks and Bayesian networks

Koelle et al. discussed the applications of Bayesian networks in social network analysis [2]. In particular, they discussed the limitations of social network analysis and the usage of Bayesian networks in social network analysis. They identified two limitations where the first one is the issues in data collection and the second one is homogeneous node and link types. For the first issue, the authors specified that there have been many sources of uncertainty in the data collection process. Therefore, by using the knowledge of these sources that will make use of the uncertain information, the validity of social network analysis can be improved. Since every person views social network differently, obtaining an objective view is difficult. Furthermore, accumulating a dataset which provides interesting conclusions will require a significant effort. For the second issue, social network does not fully address the various kind of relationships. Many traditional graph-theoretic algorithms used for social network analysis are found to be homogeneous. The three major uses of Bayesian networks in social network analysis are: reasoning about uncertainty, searching the network and inferring links. Reasoning about uncertainty means many graph theoretic algorithms do not consider the role of uncertainty and so, by using a Bayesian network, the graph theoretic algorithms can make use of the uncertainty like considering the certainty of links, the recency of links and any other type of meta information. Searching the network means by using a Bayesian network, an user can find people of similar interest in social networks. Inferring links means using a Bayesian network, new links can be deduced from the information which will be already known with different degrees of freedom.

Farine et al. made use of a Bayesian network for estimating the uncertainty and reliability of social network data [3]. The data which they were dealing with is an animal social network data. One of the main challenges they observed using that data is the limited sample size of the data which in turn gave rise to the uncertainty for estimating the rates of interaction between individuals. So, they made use of the Bayesian network to negate this problem. They found that Bayesian network gave some good information about the uncertainties in the network when the network is well sampled. But, when the sampling is too sparse, the Bayesian inferred networks will be able to come up with realistic uncertainty estimates around edge weights.

Shalforoushan et al. used Bayesian networks for predicting links in social networks [4]. They used Bayesian networks for friend recommendations. For that, the dataset which they used was soc-pokek obtained from snap.Stanford library. The dataset has 1632803 nodes and 30622564 edges. It also contains personal information about users. The dataset contains two files: relationship files and profile files. Relation files contain information like friendship considerations and the profile file contains profile features and personal attributes for each user. The main attributes that have effect on the friendship have been selected. The selected attributes are user-id, completion percentage, gender, age, region, work, education, marital status and hobbies. So, initially, they determined the attributes and similarities which has the most effect on friendship. After that, the friends who have similar interests will be given has a suggestion to each other. They also used a Friend of Friend algorithm for predicting the link. But, they found that Bayesian network performs much better in predicting an unobserved link between a pair of nodes.

1.4.2 Other techniques used for Social network analysis

Michael et al. used computationally efficient topological features for link prediction [5]. They found that the link between different users might be missing since they are not found in the online network which means that they don't have an virtual connection. They learnt that in the existing literature that the link prediction techniques are short of scalability. The major problem in link prediction techniques is the problem of extracting structural features. The authors keeping all these in mind presented a very easy way of extracting structural features for finding the missing links. Since they were able to find an easy way to extract structural features and then, they used a machine learning classifier to find the missing links. They concluded that the machine learning classifier which they designed was able to solve this problem even when applied to a complex dataset and they evaluated this model on various different social networking sites like Facebook, Instagram and Twitter.

Nesserine et al. proposed a supervised machine learning technique for link prediction in bipartite graphs [6]. The authors identified the problem of link prediction in two-mode social networks. In their research, the authors focused on two primary topics: predicting links in bipartite graph and predicting links in uni-modal graph. For doing this, the authors made use of the empirical nature of the bipartite graphs and also how they can increase the prediction of the model which are learnt. This is possible by instigating some changes to the topological features for computing the likelihood of connection of two nodes. They expressed their problem as a two class discrimination one. The authors used classic machine learning models for learning the problem of link prediction. They evaluated the model on two real-time datasets.

David et al. developed approaches for link prediction for solving the proximity of nodes in a network [7]. Their main intention was to understand what measures of proximity will lead to accurate link predictions. They used a network model to solve this problem. Therefore, when they performed experiments on the large co-authorship networks, they deciphered that the information about the future is found from the network topology alone and also some clever methods for identifying node proximity will perform better than various other direct measures.

Elaheh et al. made use of game theory and K-core decomposition for the problem of link prediction in social networks [8]. They recognized that existing link prediction techniques had problems like high time complexity, network size, sparsity and sparsity. They introduced an variation of weighted random walk based on game theory and K-core decomposition. They generated node representations using skipgram. They used Stochastic Gradient Descent (SGD) for optimization process and SGD had linear time complexity with respect to number of vertices. This improved the scalability of their model. For classifying the nodes and edges, they learnt a low dimensional representation which captures the network structure. They compared their model with state of the art techniques and evaluated based on accuracy. They found that their model has performed relatively well when compared to other models.

Yang et al. proposed a distance based model for link prediction. The authors focused on extracting users' relationship based on their mobility information [9]. They proved that for this kind of problem, Distance is going to be the primary metric. In particular, they used the distance metric to find if two people are friends by finding the distance between them. They also made use of the location metric together with distance. When they combined the information of these two metrics, they found that the distance between a user and stranger gets even larger. They proved that distance is an useful metric to solve the link prediction problem and they also used a machine learning classifier to improve the performance. They performed experiments on Twitter dataset and found that their model performs better.

1.4.3 Learning Bayesian network from the data

Dimitris Margaritis focused on the problem of determining the structure of directed models for which she used Bayesian networks [10]. The author mentioned that by learning the structure of a Bayesian network, the network will give insights into causal structure. The author also mentioned that Bayesian network is used for predicting quantities which are tough, difficult and expensive. The author proposed an algorithm for obtaining a structure of Bayesian network using statistical independent statements, a statistical test for continuous variables and an application of structure learning to a decision support where the model is learned from the data (structure).

Agnieszka et al. used the parameters of Bayesian network for learning it from small datasets [11]. This turned to be an application of Noisy-OR gates. They found that the datasets which existed now can reduce the knowledge engineering effort which is required to parameterize Bayesian network. However, they found that when the dataset is small, a lot of conditioning cases are represented by few or no data. So, they used the concept of Noisy-OR to negate this problem and by reducing the requirements of data for learning conditional probabilities. They tested their model for diagnosing liver disorders and found out that their model performs well.

Cohen et al. focused on learning Bayesian networks for facial expression recognition for both labeled and unlabeled data [12]. In particular, they used a Bayesian network classifier [12]. They found that understanding the emotions of human is a important skill since it can be used for the computer to interact intelligently with them. They created a Bayesian classifier for classifying the expressions from a video. They found out that Bayesian networks can handle missing data better for both training and testing. Their main focus however was on labeled and unlabeled data. They showed when they used unlabeled data for learning classifiers, the performance of the classifier was improved. They then introduced an stochastic structured based algorithm for learning the structure of Bayesian network. They found that the model makes use of the unlabeled data to improve the performance of classifier.

Nir et al. concentrated on learning Bayesian networks for massive datasets [13]. They labeled their algorithm has an sparse candidate one. They decoded that the standard heuristic techniques doesn't work quite well for large datasets since the search procedures spend a lot of time finding the candidates that are irrelevant. To overcome this problem. the authors designed an algorithm that achieves faster learning process by restricting the search space. This iterative algorithm limits the parameters of each variables to belong to a small subset of candidates. After this, they will search the network which will match these candidates. They evaluated the network on real time data and found that their model performs really well.

Tommi et al. negated the problem of combinatorial by finding the highest scoring Bayesian network which is learnt from the data [14]. The authors viewed this structured learning problem as an inference problem since the variables mention the choise of parents for each node in a graph. There is a global constraint that a graph has to be an acyclic one and this was the core combinatorial problem. Thus, they casted this structured learning problem as linear over a polytope. For modifying this problem, they maintained an outer bound approximation to the polytope and for searching the valid constraints, they iteratively tightened it. This will find the right Bayesian network and the results suggested that the model performed well.

1.4.4 Structured learning of Bayesian networks

Pedro et al. used structured learning of Bayesian networks for analyzing the performance of control parameters [15]. In particular, they addressed the problem of search for the Bayesian network which is best when a database of cases are given. By also using the genetic algorithm, they found the method of searching among alternative structures. For constructing the network, they started by ordering the nodes of network structures. This is needed since the networks which are chosen by genetic algorithm should be a legal one. Next using a repair operator, they convert illegal structures to legal structures. They show that the best results are obtained with an elitist genetic algorithm.

Cassio et al. tackled the problem of structured learning of Bayesian networks using constraints [16]. In particular, they addressed the exact learning of Bayesian network structure from data and also experts knowledge which is based on score functions. This will describe the properties that will lessen the time and memory costs of algorithms like hill-climbing and dynamic programming. After that, they presented a Branch and bound algorithm which will integrate both the parameters and structural constraints and these wil in turn ensure global optimality. These have the properties of being applied to large datasets and the existing methods cannot do this.

Lobna et al. found that they can improve the algorithms for structured learning in Bayesian networks using the concept of implicit score [17]. The authors investigated that in the existing research, the most commonly used heuristic search for graph is by defining a score metric and employing a search strategy to find the network which will have the maximum score. Therefore, the authors proposed a new metric called the implicit score and implemented this with the help of K2 and MWST algorithms for network structure learning. They evaluated this on a benchmark database and found that this new metric performs well.

Mikko et al. presented an algorithm which finds the most accurate posterior probability of a sub network [18]. This is actually an modified version of the algorithm which finds the most probable network structure. They found out that this exact computation will be helpful in solving complex cases where the existing methods like Monte Carlo and local search procedures fail. They also show that when a domain contains large number of variables, exact computation will be feasible given certain restrictions like priori and when both exact and inexact methods are possible.

1.4.5 Comparison of my work to the literature

When I compare my work to the literature, it has to be said that there have been no papers which even gives a small insight about my work. This is because my research involves finding a Bayesian network for a social network problem and even though there are lot of papers focused on Bayesian networks but none of those papers does not involve the problem of social networks. So, my work is a novel idea since we are using Bayesian network to solve the problem of social networks and also since we try to find the right Bayesian network structure for the same problem. The other difference is also we focus on structured learning part of Bayesian network for finding a solution for social networks and in the existing literature, there have been no papers centered upon this.

Therefore, my work is completely new due to these reasons:

- Using a Bayesian network for a social network problem
- Using structured learning in Bayesian networks for social networks
- Finding the right Bayesian network structure for social networks

1.5 Structure of the report

The report is structured as follows: Chapter 2 gives background on SCI and also on Bayesian networks. Chapter 3 provides the methodology for constructing the Bayesian network and Chapter 4 gives the results and discussion. Chapter 5 is the conclusion.

2. Background

2.1 Social connectedness index

The **Social Connectedness Index**, abbreviated to $\text{SCI}_{i,j}$, calculates the relative probability of a Facebook friendship link between a user in the location i and user in the location j. The SCI is defined as follows:

$$\text{SCI}_{i,j} = \frac{C_{i,j}}{U_i \cdot U_j} \tag{2.1}$$

In Equation (2.1), U_i and U_j depicts the total number of connections among number of users between two different locations i and j. $C_{i,j}$ depicts the total number of Facebook friendship connections between people in two different locations, i and j. If the measure is twice as large the original measure, then the person in the location i is more likely to connected to the person in the location j.

The product $U_i \cdot U_j$ gives the maximum number of connections between people at location i and j. On the other hand, $C_{i,j}$ is the number of actual connections. Hence, the SCI is the ratio between actual and maximal number of connections between people.

The data of Social Connectedness Index contains the SCI measured between two different geographical regions. Each dataset will contain i to j location pairs and vice versa. It also includes links of each location to itself. Every dataset has three columns:

- $user\_loc$- First Location
- fr_loc- Second location
- $scaled\_sci$- Scaled SCI has explained above

The datasets which are included within the folder contain the $\text{SCI}_{i,j}$ for the following areas:

1. **Country-Country** Every row is a country-country pair. They are depicted by their ISO2 codes. It excludes countries where Facebook is banned. There are 185 unique countries in total.
2. **US County-US County** Every row is a US county-US county pair. They are depicted by their FIPS code. It does not include counties which has lesser active users.
3. **US County-US Country** Every row is a US county-US country pair. Counties are depicted by their FIPS code whereas countries are depicted by their ISO2 code. It excludes countries with fewer users.
4. **GADM/NUTS GADM/NUTS** There are two more files built on the Database of Global Administrative Areas (GADM) and the European Nomenclature of Territorial Units for Statistics (NUTS) areas. It also excludes regions with fewer users.
   - **GADM _ NUTS2**: European countries are divided into NUTS2 regions. The countries that are outside Europe are divided into their GADM level 1 regions.
   - **GADM1 _ NUTS3 _ Counties**: European countries are divided into NUTS3 regions. The United States, Canada and few countries in the Asia are divided into GADM level 2 regions. The rest of the countries are divided into GADM level 1 region.

2.1.1 Social connectedness in the United States

In the United States, Facebook has played a major role for people to interact online with their friends and acquaintances [19]. People usually become friends with people with whom they knew in real life. SCI is constructed for 3,136 US counties and also between every US county and foreign country. The highest SCI is for the Los Angeles County- Los Angeles County connections. This is the region where people have the largest number of friendship connections.

When considering the San Francisco county, it is found that the people in San Francisco have more social connections with the people in the northeastern United States. When comparing the San Francisco County with the Kern county, it is found that the Kern county have significantly lower social connections with the people in the northeastern United States. The Kern county has more friendship connections to the people in West Coast and Mountain areas. This might be due to past migration patterns since many people migrated from West Coast and Mountain Areas to the Kern county. Kern county has also more friendship links to the oil-producing regions of North Dakota since Ken county is the biggest oil producing region in the United States [19].

The Social Connectedness Index is also affected by physical obstacles such as large rivers and mountain ranges. The counties with a military base have strong connections with the entire United States. Similarly, the counties with native American Reservations are strongly connected with each other. The areas which have some common functions like ski resorts, common languages have strong connections. Similarly, the regions which have African American population have more connections with people in the southern part of the United States [19].

2.1.2 Social connectedness in Europe

Europe consists of many regions and every country in Europe have their language of their own which makes it so unique and relatively different from the United States. The Social Connectedness in Europe is influenced by many factors such as language, migration patterns, political borders, religion, education and age. Two regions which have a common language are expected to have more Social Connectedness Index than two regions which doesn't have a common language [20].

In relation with the migration patterns, people of South-West Oltenia in Romania have more connections with people throughout Europe especially in Italy, Spain, Germany and the United Kingdom. This is due to past migration history. In relation to the language, Limburg, a region in Belgium has more friendship connections with the Netherlands since in Limburg, the official language is Dutch. The regions of Slovenia, Croatia, Serbia, North Macedonia, and Montenegro are made into one community since before the division they were known together as Yugoslavia. This holds for Czech Republic and Slovakia also. Since Belgium has three official languages (Dutch, German and French), the French speaking part of Belgium is expected to have more connections with people in France [20].

2.2 Graph theory

2.2.1 Basic concepts

A **graph** G is defined as a pairs $G = (V(G), E(G))$, where $V(G)$ is a finite set of **vertices**, where a vertex is often denoted by a letter, e.g. $v, u \in V(G)$, and $E(G) \subseteq V(G) \times V(G)$ are called **edges**. We also use letters with indices, e.g. v_1, v_2 to denote vertices, or sometimes just integers.

Different types of graph can be distinguished. If we have that if $(u, v) \in E(G)$ it follows that $(v, u) \in E(G)$, the graph is called **undirected**. In undirected graphs an edge (u, v) is often denoted by a **line**, or (undirected) edge $u - v$. On the other hand, if it holds that if $(u, v) \in E(G)$ then $(v, u) \notin E(G)$ the graph is called **directed**. An edge $(u, v) \in E(G)$ in a directed graph G is often indicated by $u \to v$, called an **arc** or directed edge.

Let G be an undirected graph, in that case it is possible to travel through the graph from vertex to vertex, just by following the vertices that are connected to each other by lines, $v_1 - v_2 - \cdots - v_p$, called a **path**. If a path between two vertices v_1 and v_p exist in graph G, they are said to be

connected. A similar situation exists for a directed graph G, but here paths can have different forms, e.g. the form of a **directed path** $v_1 \to v_2 \to \cdots \to v_p$. If for a directed graph G it holds that it does **not** contain a path of the form $v_1 \to v_2 \to \cdots \to v_p = v_1$, called a **directed cycle**, it is called **acyclic**.

In particular in social network analysis, special graphs are used where lines or arcs have an attached weight w; these graphs are called **weighted graphs**, formally denoted as $G = (V, E, w)$, where $w : E(G) \to \mathbb{R}$ acts as a **weight function**.

2.2.2 Some further concepts

For directed acyclic graphs, DAG for short, there is some special terminology used. Let $G = (V(G), E(G))$ be a DAG, then if $u \to v \in E(G)$, then u is called the **parent** of v, whereas v is known as the **child** of u. If a vertex u can be reached from vertex v by a directed path starting from v, then u is also known as a **descendant** of v. Note that a child is a descendant of its parent node; the concept of descendant allows for describing children of children. Furthermore, often when considering paths in a DAG we are not always interested in the direction of the edges connecting the vertices on the paths. In that case, we ignore the direction of edges on the path by considering the **undirected version** of the DAG G, also known as the **underlying graph**.

Some other concepts used in the following are:

- Vertices v, u that are connected by an edge $e = (v, u)$ are called **adjacent** and **incident** to the edge e.
- A vertex u that is not connected to any other vertex by an edge is called **isolated**.
- An **ancestor** u of a vertex v is a vertex on a directed path starting from u and ending at v, $u \neq v$.

2.3 Probabilistic graphical models

We continue with a brief review of some key concepts from probability theory, i.e., we consider events, joint probability distributions, conditional probability distributions, the chain rule, marginalization, and conditional independence, after which we describe probabilistic graphical models.

2.3.1 Basic probability concepts

Let $X = \{X_1, \ldots, X_n\}$ be a set of random variables, where $D(X_i)$ indicates the **domain** of variable $X_i \in X$. The domain of X is the Cartesian product $D(X) = \times_{i=1}^{n} D(X_i)$. An (elementary) **event** $E \equiv X = x$ is any random variable X with a value x from its domain. The set of all possible Boolean combinations of events, or **Boolean algebra** denoted as $\mathcal{B}(X)$, is defined by using the operators: conjunction $(X = x \wedge X' = x')$, disjunction $(X = x \vee X' = x')$, and negation $(\overline{X = x})$. This Boolean algebra contains events such as $(X_1 = x_1 \vee X_2 = x_2)$, $(X_3 = x_3 \wedge X_4 = x_4)$, and $\overline{X_2 = x_2}$. Events are partially ordered by $\leq$, with the universal lowerbound $\perp \in \mathcal{B}(X)$ and universal upperbound $\top \in \mathcal{B}(X)$, i.e., we have for each $E \in \mathcal{B}(X)$ that $\perp \leq E$ and $E \leq \top$. Usually $(X = x \wedge X' = x')$ is represented in set notation as $\{X = x, X' = x'\}$. Note that often we do not make a distinction between elementary events, i.e. $X_1 = x_1$ and a conjunction or sets of events, i.e. $X = x$ which might stand for $(X_1, X_2) = (x_1, x_2)$ or $\{X_1 = x_1, X_2 = x_2\}$.

A probability distribution is a function or mapping that assigns probabilities, i.e., values from the closed real interval $[0, 1]$, to any event involving variables in X.

Definition 1 (Probability distribution). *A probability distribution for a set of random variables X with* ***domain*** *$D(X)$ is defined as a function $P : \mathcal{B}(X) \to [0, 1]$, such that the following axioms hold:*

(1) $P(E)$ is a non-negative real value for all $E \in \mathcal{B}(X)$;

(2) $P(\top) = 1$;

(3) for any set of disjoint events $E_1, \ldots, E_n \in \mathcal{B}(X)$, with $(E_i \wedge E_j) = \bot$, $1 \leq i, j \leq n$, $i \neq j$, we have that:

$$P\left(\bigvee_{k=1}^{n} E_k\right) = \sum_{k=1}^{n} P(E_k).$$

It is a fundamental property of probability theory that it is sufficient to specify a probability distribution in terms of joint events $\{X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\}$, i.e., in terms of a **joint probability distribution** $P(X_1, X_2, \ldots, X_n)$ for all values of the domain $D(X)$ (possibly with the exception of one element from $D(X)$, where its probability can be derived from the other probabilities of elements of $D(X)$ according to axioms (2) and (3)).

When the actual value of a random variable in an elementary event does not matter in a given context, we often also write $P(X)$ rather than $P(X = x)$ for the probability of variable X taking the value x.

The **marginal probability distribution** for a set of variables Y given the probability distribution for the random variables X, with $Y \subseteq X$ and $X = Y \cup Z$, where Y and Z are disjoint, is obtained by summing out the other variables (i.e. Z) from the joint probability distribution $P(X)$, and is defined as:

$$P(Y) = \sum_{z \in D(X \setminus Y)} P(Y, Z = z)$$

Let $P(X, Y)$ be a joint probability distribution over a set of random variables X and Y. A **conditional probability distribution** $P(X \mid Y)$ is defined as:

$$P(X \mid Y) = \frac{P(X, Y)}{P(Y)} \tag{2.2}$$

with $P(Y) > 0$.

It is good to realize that $P(X \mid Y)$ is actually a family of probability distributions, one for every value y of Y. The conditional probability $P(X = x \mid Y = y)$ is the probability of the event $X = x$ given knowledge about the event $Y = y$.

The concept of conditional probability is one of the most fundamental and most important concepts in probability theory. In addition, the conditional probability plays an essential role in a wide range of domains, including classification, decision making, prediction and other similar situations, where the results of interest are based on available knowledge.

By moving the denominator on the right of Equation 2.2 to the left, Equation 2.2 can also be written as:

$$P(X, Y) = P(X \mid Y)P(Y) = P(Y \mid X)P(X) \tag{2.3}$$

By applying Equation 2.3 to a set of random variables $\{X_1, X_2, \ldots, X_n\}$, this creates a chain of conditional probabilities, more formally:

Proposition 1 (Chain Rule)**.** *Let P be a joint probability distribution over a set of random variables $X = \{X_1, X_2, \ldots, X_n\}$. Then it holds that:*

$$P(X_1, X_2, \ldots, X_n) = P(X_n \mid X_{n-1}, \ldots, X_1) \cdots P(X_2 \mid X_1)P(X_1) \tag{2.4}$$

The chain rule allows us to compute the joint distribution of a set of any random variables by only making use of conditional probabilities. This rule is particularly useful in Bayesian networks, which we will introduce later in this chapter. Combined with the network structures, the use of the chain rule can facilitate the representation for a joint distribution.

Another immediate result of Equation 2.3 by rearranging terms is **Bayes' rule**, also known as **Bayes' theorem**:

$$P(X \mid Y) = \frac{P(X)P(Y \mid X)}{P(Y)} \tag{2.5}$$

Bayes' rule tells us how we can calculate a conditional probability given its inverse conditional probability. For example, using Bayes' rule makes it possible for us to derive the conditional probability $P(X \mid Y)$ from its inverse conditional probability $P(Y \mid X)$, if we also have information about the prior probability $P(X)$, $P(Y)$ of events X and Y respectively. $P(Y)$ also behaves as a normalizing constant.

A more general conditional version of Bayes' rule, where all probabilities are conditional on the same set of variables Z, also holds:

$$P(X \mid Y, Z) = \frac{P(X \mid Z)P(Y \mid X, Z)}{P(Y \mid Z)}$$

with $P(Y \mid Z) > 0$.

Another fundamental concept in probability theory is **conditional independence**. Two sets of variables X, Y are said to be conditionally independent given a set of variable Z, denoted $X \perp\!\!\!\perp_P Y \mid Z$, if

$$P(X \mid Y, Z) = P(X \mid Z) \quad \text{or} \quad P(Y, Z) = 0 \tag{2.6}$$

Equation 2.6 asserts that given knowledge of a set of variables Z, knowledge of whether Y occurs provides no extra information on the probability of whether X occurs.

2.3.2 Bayesian networks

Bayesian Networks, which are also known as belief networks or Bayes nets, are one of the types of probabilistic graphical models. A Bayesian Network (BN) is used for representing a joint probability distribution, taking into account the conditional independence's that hold among the random variables involved in the distribution. Bayesian network is an I-map (Independence map). A G is called an undirected I-map if the following holds: $X \perp\!\!\!\perp_G Y|Z \implies X \perp\!\!\!\perp Y|Z$.

A more formal definition of Bayesian Network can be given:

Definition 2 (Bayesian network)**.** *A Bayesian network $\mathcal{B}$ is defined as a pair $\mathcal{B} = (G, \Theta)$ where G is a DAG with vertices $V(G) = \{1, 2, \ldots, n\}$,corresponding 1–1 to random variables $X = \{X_1, X_2, \ldots, X_n\}$, and arcs $E(G) \subseteq V(G) \times V(G)$, representing probabilistic independence information; Θ denotes the probabilistic parameters of the network. The following holds: $\Theta_{i|\pi(i)} = P_{\mathcal{B}}(X_i = x_i \mid X_{\pi(i)} = x_{\pi(i)})$ for each realisation $X_i = x_i$, conditioned on the values of the set of parents $X_\pi(i) = x_{\pi(i)}$. Correspondingly, $\mathcal{B}$ defines a unique joint probability distribution (JPD) on the random variables X using the chain rule:*

$$P_B(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \prod_{i=1}^{n} P_B(X_i = x_i|X_{\pi(i)} = x_{\pi(i)}) = \prod_{i=1}^{n} \Theta(i \mid \pi(i)) \tag{2.7}$$

In the following we will no longer explicitly distinguish between vertices i and the corresponding variable X_i of a Bayesian network.

Hence, a Bayesian network is just a factorisation of a JPD, using the chain rule 2.4, taking into account the conditional independence imposed by the graph structure of the network. This is a consequence of the **Markov condition**: any variable is conditionally independence of non-descendants given its parents. The relationship between the structure of the graph of a Bayesian network and the conditional independence's that follow from these is not straightforward and are described by the concept of **d-separation**.

When vertex Y has converging arcs (head-to-head) $\cdot \rightarrow Y \leftarrow \cdot$, it is called a **collider**. The vertices with other arc connections are called non-colliders and they connect to other vertices as follows:

- tail-to-tail arcs: $\cdot \leftarrow Y \rightarrow \cdot$
- tail-to-head arcs: $\cdot \leftarrow Y \leftarrow \cdot$ or $\cdot \rightarrow Y \rightarrow \cdot$
- head-to-head arcs: $\cdot \rightarrow Y \leftarrow \cdot$

Later it will appear that head-to-tail is equivalent to tail-to-head, which is why we do not distinguish it. D-separation is defined in terms of blocking.

Notion of Blocking If there are three subsequent variables X_i, X_j, and X_k on an undirected path, the path passing through X_i, X_k, and X_j will be called

- **blocked** by X_k if X_i and X_j are connected by a path in the underlying graph where X_k is the middle vertex and X_k is connected to X_i and X_j as a non-collider, i.e. tail-to-tail or tail-to-head;
- **blocked** (by nothing) if (X_i, X_k, X_j) form a collider, and **unblocked** if in the collider (X_i, X_k, X_j) X_k or one of its descendants are given.

D-separation Next, the formal definition of d-separation is given:

Definition 3. *Let $G = (V(G), E(G))$ be a DAG with $X, Y, Z \subseteq V(G)$ sets of vertices. If every path from a vertex in X to every vertex in Y is blocked by a vertex in Z we say that X and Y are **d-separated** given Z written as $X \perp\!\!\!\perp^d_G Y \mid Z$.*

If vertices X and Y are not d-separated given Z, we say that they are **d-connected**, written as $X \not\perp\!\!\!\perp^d_G Y \mid Z$.

Figure 2.1 show an example of a DAG.

The tail-to-tail arc in Figure 2.1 is $u \leftarrow v \rightarrow y$. One of the tail-to-head arcs is $\leftarrow u \leftarrow$. Head-to-head arc in the figure is $s \rightarrow t \leftarrow u$.

In Figure 2.1, there is a path between s and v, but since t is a collider, s and v are d-separated. This is also true for the path $x - r ... - y$, where x and y are d-separated for the same reason. But, if we consider the path but now given t, s and v would be d-connected. x and y are also d-connected since the path is unblocked by t.

Let us assume that Z is a set and contains a set of variables r, v. So, then x and y are d-separated by Z and the same applies for x and s, u and y, s and u. The paths $x - r - s$, $u - v - y$ and $s - t - u$ is blocked by Z. The only pairs of nodes which remain d-connected conditioned on Z are s and t, u and t. Even though t is not in z, the path is nevertheless blocked by Z since t is a collider.

2.3.3 Markov Independence

Independence relation

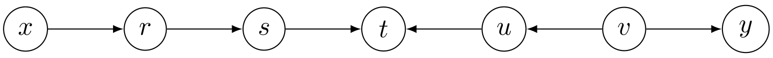


Figure 2.1: D-separation

Definition 4. *If X, Y, $Z \subseteq V$ are sets of random variables and if P is a probability distribution of V then X is called as conditional independent of Y given Z, denoted as*

$X \perp\!\!\!\perp_P Y|Z$, if and only if $P(X|Y,Z) = P(X|Z)$

The $\perp\!\!\!\perp_P$ relation The relation $X \perp\!\!\!\perp_P Y|Z$ defines a ternary predicate $\perp\!\!\!\perp_P (X,Y,Z)$. It also holds the symmetry property which is denoted as:

$$X \perp\!\!\!\perp_P Y|Z \Longleftrightarrow Y \perp\!\!\!\perp_P X|Z \tag{2.8}$$

Properties of the $\perp\!\!\!\perp_P$ relation

- **Symmetry:** If Y gives no information about X given Z, then X won't give any extra information about Y. If $X, Y, Z \subseteq V$ are set of variables, then:

$$X \perp\!\!\!\perp_P Y|Z \Longleftrightarrow Y \perp\!\!\!\perp_P X|Z \tag{2.9}$$

- **Contraction:** If Y is unrelated to X given Z and if W is judged to be unrelated to X after hearing the information about Y, then W should have been irrelevant prior to learning Y. If $X, Y, W, Z \subseteq V$ are disjoint sets of random variables, then:

$$X \perp\!\!\!\perp_P Y|Z \wedge X \perp\!\!\!\perp_P W|Y \cup Z \implies X \perp\!\!\!\perp_P W \cup Y|Z \tag{2.10}$$

Global Markov property-separation

Definition 5. *If $G = (V(G), E(G))$ is an undirected graph and if $U, W, Z \subseteq V(G)$ are vertices in G, the set W (u)-separates U and Z denoted as, $U \perp\!\!\!\perp_G Z|W$ if every path from a vertex in U to a vertex in Z contains at least one vertex in W or else these sets are u-connected.*

Markov Network A Markov network is a set of random networks having a Markov property which is described by an undirected graph. A Markov network is similar to a Bayesian network in the representation of its dependencies. The difference between a Bayesian network and a Markov network is that Bayesian networks are directed and acyclic whereas Markov networks are undirected and might be cyclic. Due to this, Markov networks can represent certain dependencies which Bayesian networks cannot.

2.3.4 Conditional Independence

Two random events X and Y are conditionally independent given a third event Z precisely if the occurrence of X and Y are independent events in their conditional probability distribution given Z. Conditional independence is basically the concept of independence $P(X \cap Y) = P(X) * P(Y)$ applied to a conditional model.

The mathematical definition of conditional independence is given:

Definition 6. *In probability theory, X and Y are conditionally independent given Z if and only if $P(X \cap Y|Z) = P(X|Z)P(Y|Z))$. Conditional independence of X and Y given Z is denoted by $(X \perp\!\!\!\perp Y)|Z$. Formally,*

$$(X \perp\!\!\!\perp Y)|Z \iff P(X \cap Y|Z) = P(X|Z)P(Y|Z) \tag{2.11}$$

or equivalently,

$$(X \perp\!\!\!\perp Y)|Z \iff P(X|Y \cap Z) = P(X|Z) or P(Y|Z) = 1 \tag{2.12}$$

2.3.5 Example of a Bayesian network

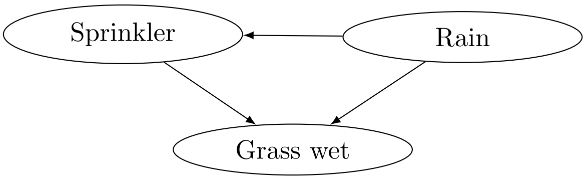
Figure 2.2 shows an example of a Bayesian network model which describes the relationship between the season whether rain falls during the cloudy season, whether the sprinkler is on during the cloudy season and whether the grass would get wet. Every node is specified by a Conditional Probability Distribution (CPD). If the variables are discrete, the distribution can be represented in the form of a table called the Conditional Probability Table (CPT) which will contain the combination of values of nodes parents the probability that the node takes on each of its different values [21].

Rain: When the season is cloudy, the probability of rain is 0.8 otherwise the probability of rain is 0.2.

Sprinkler: When the season is dry, the probability that the sprinkler is on is 0.4 and the probability that the sprinkler is off is 0.6. When there is rain, the probability of sprinkler being on is 0.01 and the probability of sprinkler being off is 0.99.

Grass Wet: For the Grass Wet node, there are two possible causes: the probability that the sprinkler is on or it is raining.

	Sprinkler	
Rain	*T*	*F*
F	0.4	0.6
T	0.01	0.99



Sprinkler	
T	*F*
0.2	0.8

		Grass wet	
Sprinkler	Rain	*T*	*F*
F	*F*	0.4	0.6
F	*T*	0.01	0.99
T	*F*	0.01	0.99
T	*T*	0.01	0.99

Figure 2.2: Example of Bayesian Network

2.3.6 Special form Bayesian Networks

The two special kind of Bayesian networks are:

- naive (independent) form Bayesian networks
- Tree-Augmented Bayesian networks

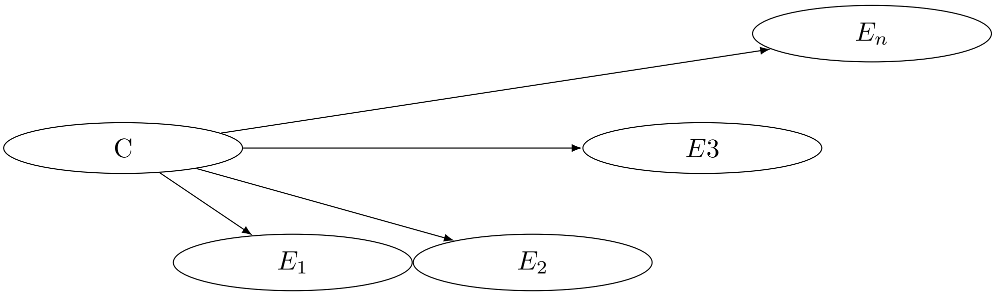


Figure 2.3: Naive form Bayesian network

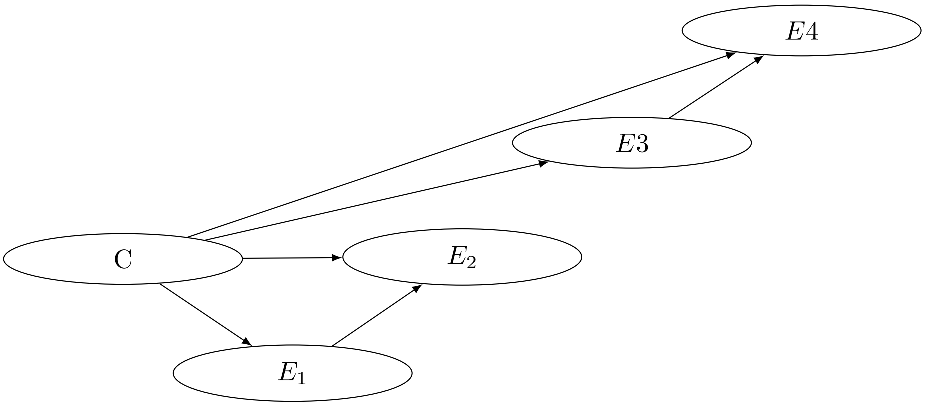


Figure 2.4: Tree-Augmented Bayesian Network

Naive form Bayesian network

Definition 7. *If C is a class variable and if E_i are the evidence variables and $\mathcal{E} \subseteq \{E_1, \ldots, E_m\}$. It is assumed that $E_i \perp\!\!\!\perp E_j \mid C$. So, by Bayes$'$ rule:*

$$P(C|\mathcal{E}) = \frac{P(\mathcal{E}|\mathcal{C})P(C)}{P(\mathcal{E})}$$

with:

$$P(\mathcal{E}|\mathcal{C}) = \prod_{E \in \mathcal{E}} P(E|C)$$

by conditional independence

$$P(\mathcal{E}) = \sum_{C} P(\mathcal{E}|\mathcal{C})P(C)$$

Tree-Augmented Bayesian Network (TAN) TAN is the extension of Bayesian network which is used for reducing the number of independent assumptions. As with naive Bayes you have a class variable and evidence variables, and an evidence variable has two parents: the class variable and another evidence variable. Together the evidence variable form a tree. This does indeed allow representing dependence's between evidence variable (not always via the class variable as holds for naive Bayes).

2.3.7 Description about probability distributions

Continuous probability distributions A continuous probability distribution is a distribution where the random variable X can take on any value that is continuous. Since X can assume infinite values, the probability of X taking on one specific value is 0. The continuous random variable is infinite and uncountable.

Discrete probability distributions A discrete probability distribution is a distribution which describes the set of possible outcomes in a discrete way (eg:coin toss, roll of a die). In a discrete probability distribution, the probabilities are encoded by a discrete list of probabilities of outcomes which is called as the Probability Mass Function. A discrete probability distribution is countable.

Mixed Random Variables If a random variable is neither continuous nor discrete, then that random variable is called as a mixed random variable. The mixed random variable has both the continous and discrete part.

Multivariate Gaussian distribution A Gaussian distribution $X \sim \mathcal{N}(\mu, \sigma^2)$ is also called as Normal distribution. It is kind of continuous probability distribution for real valued random variable. The probability density function of Gaussian distribution is:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{2.13}$$

The parameter μ is called the mean or expectation of the distribution. The parameter σ is called the standard deviation and the variance of the distribution is σ^2. A random variable with a Gaussian distribution is said to be normally distributed and is called a normal deviate.

A multivariate Gaussian distribution is a generalization of the univariate Gaussian distribution to higher dimensions. It is often used to describe a set of correlated real values random values clustered around a mean.

$$\begin{aligned} f(y_i|\mu, \Sigma) &= \\ &= f((y_{i1}, y_{i2}, ..., y_{in})' \,|\, \mu = (\mu_1, ..., \mu_n)', \Sigma = \begin{bmatrix} \sigma_1^2 & & \sigma_{1n} \\ & \ddots & \\ \sigma_{n1} & & \sigma_n^2 \end{bmatrix}) \\ &= \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} exp(-\frac{1}{2}(y_i - \mu)'\Sigma^{-1}(y_i - \mu)) \end{aligned} \tag{2.14}$$

The symbol $|\Sigma|$ refers to the determinant of the matrix Σ. The determinant is a single real number. The symbol Σ^{-1} is the inverse of Σ, a matrix for which $\Sigma\Sigma^{-1} = I$. This equation assumes that Σ can be inverted, and one sufficient condition for the existence of an inverse is that the determinant is not 0.

The matrix Σ must be positive semi-definite in order to assure that the most likely point is $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ and that, as y_i moves away from μ in any direction, then the probability of observing y_i declines.

The denominator in the formula for the multivariate normal distribution is a normalizing constant, one which assures us that the distribution integrates to 1.0.

Multinomial distribution The Binomial is actually the Bernoulli distribution. The Binomial is Bernoulli applied to a sequence of tests. The Binomial distribution with parameters n and p is a discrete type probability distribution where it is used for sequence of n independent experiments where each experiment should answer a yes or no question and each of the experiments will have a Boolean valued outcome with probability p for success and probability $p-1$ for failure. A single experiments will be called a Bernoulli experiment where the sequence of the experiments is called a Bernoulli process for a single trial. Therefore, the Binomial distribution is a Bernoulli distribution. The notation, expectation, variance, Probability Mass Function (PMF) of Binomial distribution is given below:

$$\begin{aligned} \text{Notation:} \quad & X \sim \mathsf{Bin}(n, \theta) \\ & \text{where } n > 0 \text{ and } 0 \leq \theta \leq 1 \\ \text{Support:} \quad & \{0, 1, \ldots, n\} \\ \text{PMF:} \quad & \tbinom{n}{x} \theta^x (1-\theta)^{n-x} \\ \text{Expectation:} \quad & n\theta \\ \text{Variance:} \quad & n\theta(1-\theta) \end{aligned}$$

The **Multinomial distribution** is a generalization of binomial distribution.

The notation, expectation, variance, Probability Mass Function (PMF) of Multinomial distribution is given below:

$$\begin{aligned} \text{Notation:} \quad & X \sim \mathsf{Mult}(n, \theta) \\ & \text{where } n > 0,\ \theta = [\theta_1, \ldots, \theta_k] \text{ and } \textstyle\sum \theta_j = 1 \\ \text{Support:} \quad & \{x_1, \ldots, x_k \mid \textstyle\sum x_j = n, x_j \geq 0\} \\ \text{PMF:} \quad & \tbinom{n}{x_1, \ldots, x_k} \textstyle\prod \theta_j^{x_j} \\ \mathbb{E}[X] = \quad & n\theta \\ \mathsf{Var}[X_j] = \quad & n\theta_j(1-\theta_j) \\ \mathsf{Cov}[X_i, X_j] = \quad & -n\theta_i\theta_j \end{aligned}$$

Example: Let X denote the vector of the number of times each side of a i sided die has landed face up in n tosses of the die, and let θ_j be the probability that the number j is rolled on each toss of the die. For example, x_1 represents the number of times a '1' was rolled, x_2 represents the number of times a '2' was rolled, etc.

In this example the categories are $1, \ldots, k$, but in general they can be completely arbitrary. There is no need to make them ordered, or even numeric.

2.3.8 Learning Bayesian networks from data

In various different applications, the resulted Bayesian network needs to be determined by the use of the dataset. This will require construction of graph representation, such that someone already has knowledge and data at their disposal. After this, parameter estimation of the joint probability distribution in the Bayesian network takes place. This is called fitting Bayesian network to the data. The construction of graph in the absence of expert knowledge is done using appropriate structured learning algorithms.

So, the task of learning a Bayesian networks can be divided into two types:

- Structural learning
- Parameter learning

Table 2.1: Different cases in learning BN

Structures	Observability	Learning Method
Known	Fully known	Maximum-Likelihood estimation
Known	Partially known	Expectation-Maximization (EM) and Markov Chain Monte Carlo (MCMC)
Unknown	Fully known	Search model space
Unknown	Partially known	EM and Search model space

2.3.9 Structured learning

Structural learning identifies the topology of Bayesian network. The idea behind structured learning to find the best network structure is to score all the possible DAGs by a scoring function and choose the DAG that has the best score. Structured learning problem is NP-hard.

As seen in the Table 2.2, it will become impossible for searching the space of DAGs exhaustively in a sensible time for values of $n \geq 6$ and so, heuristic methods are needed to find the optimal network structures.

The problem of structured learning can be divided into two types. They are:

- Constraint based methods
- Search and score methods

2.3.10 Search and score method

Search and score is used for finding the quality measures of the Bayesian networks. As the name implies, this method has two types:

- Scoring metric: Used for computing the quality of Bayesian networks
- Searching procedure: Used for determining which network is the best.

2.3.11 Scoring function

Scoring function is used for measuring how well a network structure fits a data. So, it calculates the probability of network graph G given dataset D, $P(G|D)$. This scoring function should be similar which means it should return the same score for Markov equivalent DAGs. So, if D is a dataset, and if $B = (G, P)$ and $B^{'} = (G^{'}, P^{'})$ are two Bayesian networks,

$$q = \frac{P(G|D)}{P(G'|D)} \tag{2.15}$$

Where q is a Bayesian measure and Pr is the probability distribution used for ranking the Bayesian network structures. It has to be noted that,

$$q = \frac{P(G|D)/P(D)}{P(G'|D)/P(D)} = \frac{P(G,D)}{P(G',D)} \tag{2.16}$$

and

$$P(G,D) = P(D|G)P(G) \tag{2.17}$$

So,

$$\log P(G,D) = \log P(D|G) + \log P(G) \tag{2.18}$$

Table 2.2: Number of possible DAGs for different number of variables

Number of variables n	Number of possible DAGs
1	1
2	3
3	25
4	543
5	29,281
6	3,781,503
7	1,138,779,265
8	78,370,2329,343
9	1,213,442,454,842,881
10	4,175,098,976,430,598,100

which must be found for every Bayesian network B. For determining $P(D|G)$, three assumptions must be made: no values should be missed in D, the case $v \in D$, where v signifies vertices, should have occurred independently and there should be discrete network parameters. So, the quality measure of a Bayesian network is:

$$P(D|G) = \prod_{i=1}^{N} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk} n_{ijk}$$

where N is the number of variables, q_i represents the number of states over the parents X_i, r_i represents the number of states of X_i, θ is the estimate of the model and n_{ijk} represents the number of cases in the database with X_i in its kth state and the parent of X_i in its jth state. This measure is used for determining the maximum likelihood parameters of the model.

2.3.12 Search algorithms

The main idea behind search algorithm is to find the most probable network structure given the dataset. The algorithms which are described in this section have various different ways of searching for this structure.

Exhaustive search

Exhaustive search is also known as brute force search. It does not reduce the search space of possible DAGs instead it generates all possible DAGs and chooses the one with highest score. The DAG whose score is the highest is the global optimum, so this structure is the best possible structure.

Greedy search

Greedy search is a simple search algorithm. The algorithm will start with a initial network structure G. For each step, the algorithm defines a set of neighbourhood graph and calculates the score of each graph in this neighbour set. The neighbour graph which has the highest score is selected and used for the next iteration. The search is stopped where there is no neighbourhood network graph with a score higher than the current structure.

Tabu search is a form of greedy search with some extras. Local search methods grasps a likely solution and checks its immediate neighbours in order to find a better solution. But, local search methods gets stuck in suboptimal regions where there are lot of solutions which are equally better.

By using Tabu search, we can rectify this. Tabu search improves this by modifying the first rule. So, initially, if none of the better solutions are found, worsening moves will be preferred. Furthermore, 'prohibitions' are initiated so that the search method doesn't again search the previously visited solutions.

2.3.13 Constraint based structured learning

Constraint based structured learning is used for determining conditional independencies. They approach the problem by various statistical conditional independence tests in order to produce the dependencies between the variables. DAG is used for illustrating the dependencies and independencies. The algorithms is used for calculating the conditional independencies between the variables. After that, the constraints of conditional independencies are then spread across the DAG. The next step is to eliminate the incompatible ones. These algorithms produce only I-equivalent graphs that are the ones with which specify identical independence relations. The main basis of Constraint based algorithm is the Pearl's Inductive Causation algorithm. The most commonly used constraint based algorithms is the PC algorithm.

PC algorithm

The PC algorithm involves the following steps:

- Initially, the conditional independencies of variables are tested to derive the conditional dependencies and independencies of variables.
- The graph skeleton (=undirected graph) is identified induced by those relations.
- The convergent ($X \rightarrow Z \leftarrow Y$ structures) and divergent connections are identified.

2.3.14 Parameter learning

Parameter learning is a method where the model is fit to the data by producing an estimation of the parameters of the global parameter distribution. If a given structure is known to the user, either through proper structure learning algorithm or prior knowledge, we calculate the estimation of parameters of local distribution. So, every node will have a corresponding CPT. This will reflect the node's CPD, according to the values of the parent nodes.

2.4 Evaluation metrics

So, here we will introduce the 5 metrics which we will use for evaluating the Bayesian network.

2.4.1 Confusion matrix

Confusion matrix is a table which describes the performance of classification model. The confusion matrix reports number of False Positives (FP), False Negatives (FN), True Positives (TP) and True Negatives (TN). True Positive is a case where the model will exactly predict the positive class. True Negative is an outcome where the model will accurately predict the negative class. False Positive is a case where the model won't predict the positive class correctly. False Negative is an outcome where the model won't predict the negative class correctly. Accuracy is the ratio of observations which we correctly predict to total number of observations.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{2.19}$$

2.4.2 Receiver Operating Characteristic (ROC) curve

To evaluate a classification model at all the thresholds of classification, a ROC curve is used. ROC curve is useful for plotting two parameters: True Positive Rate (TPR) and False Positive Rate (FPR).

TPR is otherwise known as recall. The formula for TPR is

$$TPR = \frac{TP}{TP + FN} \tag{2.20}$$

The formula for FPR is

$$FPR = \frac{FP}{FP + TN} \tag{2.21}$$

So, the ROC is curve will plot FPR vs TPR for all classification thresholds. If any curve is close to the top left corner, then it will indicate that the model has performed well.

2.4.3 Brier score

Brier Score (BS) is an another way of verifying if a probabilistic model has performed well. Brier score is quite similar to the Mean Squared Error.

The formula of Brier score is

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2 \tag{2.22}$$

where f_t is the predicted probability and o_t is the actual outcome of the event and N is the number of total observations.

2.4.4 Log-likelihood

To find the goodness of fit of a probabilistic model, the likelihood can used. The **likelihood** l is just the probability of the data given the model: $P(D : M)$, where D are the data and M the model and P is obtained from the probabilistic model M. Usually it is assumed that the records in the data D are independent and identically distributed (iid, for short) (using M) and this allows to compute

$$l(D : M) = P(D : M) = \prod_{r \in D} P(r : M)$$

However, as it is usually easier to compute a sum, rather than a product, usually the **log-likelihood** L is computed:

$$L(D : M) = \log P(D|M) = \sum_{r \in D} \log P(r : M)$$

A disadvantage of the log-likelihood is that when the probability $\downarrow 0$, the log-likelihood $\to -\infty$.

2.4.5 Bayesian Information Criterion (BIC) and Bayesian Dirichlet equivalence (BDe) score

BIC which is also known as Schwarz Information Criterion is used for scoring and choosing a model. The BIC is defined as

$$BIC = k \log n - 2\hat{L}(D : M) \tag{2.23}$$

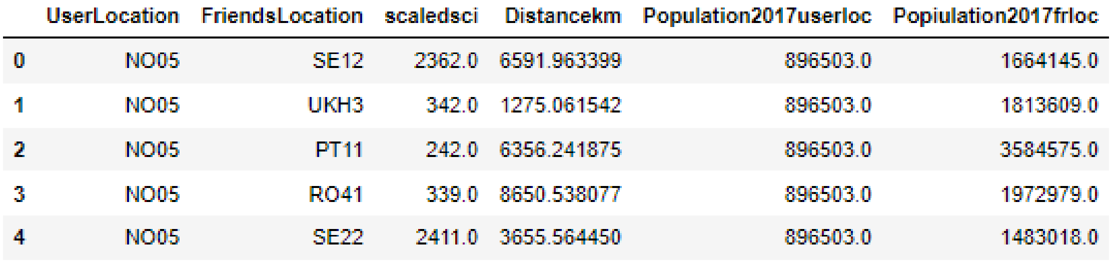
where

- $\hat{L}$ is the largest value of the log-likelihood function of the model
- n are the number of observations
- k are the number of parameters estimated by the model M

BDe is an another scoring metric used for computing the score of a Bayesian network. Dirichlet process belong to a family of stochastic process who realizations are expressed in probability distributions. They are very common in Bayesian networks where they are used for describing the prior knowledge about how the random variables are distributed which means finding out how likely the random variables are distributed according to one or more distributions.

3. Methodology

3.1 Description of the dataset



	UserLocation	FriendsLocation	scaledsci	Distancekm	Population2017userloc	Popiulation2017frloc
0	NO05	SE12	2362.0	6591.963399	896503.0	1664145.0
1	NO05	UKH3	342.0	1275.061542	896503.0	1813609.0
2	NO05	PT11	242.0	6356.241875	896503.0	3584575.0
3	NO05	RO41	339.0	8650.538077	896503.0	1972979.0
4	NO05	SE22	2411.0	3655.564450	896503.0	1483018.0

Figure 3.1: Description of the dataset

The figure for the description of the dataset can be seen in 3.1. The dataset contains information about a person and the person's friend living in two different locations which is represented in the dataset by Userlocation and Friendslocation. The scaled SCI represents the strength between people living in two different regions. The distance represents the distance between the two regions (userlocation and friendslocation). The population reflects upon the number of people living in the regions. Furthermore, every country are divided into many regions and all those regions are uniquely identified by their codes. So, a country is not represented as a whole

3.2 Description of the features

3.2.1 Input variables

User Location and Friends Location

For both User Location and Friends Location, first initially in the data, all the countries are divided into various regions and every region in the country are represented by a unique code. So, we remove all these unique codes and consider every country as one. After this, we consider only a small subset of countries. We also use the population feature for obtaining the overall population of every country. This will help us in obtaining the prior probabilities for every country. That is the main reason why the prior probabilities for both the user location and friends location are similar.

Population

The population feature represents the number of people living in both the pop_user and pop_fr location. This feature is continuous. We use the population feature for calculating the prior probabilities of user location and friends location.

dist_km

The distance feature represents the distance between the location of i and j. This feature is also continuous. We convert them to discrete by using intervals of discretization.

Using countries as features

We also use countries as features for performing our analysis. We also consider only a small set of the countries and since earlier we didn't have prior probabilities in the data, we will use the population feature to obtain the prior probabilities for every country.

3.2.2 Target variable

scaled_sci

This feature is used for calculating the strength of social connectedness between two regions. This feature is continuous. We convert them to discrete by using intervals of discretization.

3.3 Aim

The main thing which we are going to do is to find the right Bayesian network structure for this problem. We will also see the probability distribution of the features. Therefore, we will use a Naive bayes network and also a search based network in the form of Tabu search and also compare these two networks.

So, the input and the output variables which we will be using are described above. Regarding the input variables, as mentioned above, we will use the population feature for calculating the prior probabilities of user location and friends location and it will also help us in obtaining the total population of every country. instead it is divided into many regions and those regions are indexed by a code.

3.4 Construction of Naive Bayes Network

We construct Naive Bayes network and predict scaled sci based on distance and population features. Thus, to achieve this, we convert continuous variables to discrete variables using quantiles.

3.4.1 Quantiles

Quantiles are cut points which is used for determining how many values are above or below certain limit. It is used for cutting the distribution into equally sized, adjacent subgroups. In our case, it is used for cutting the distribution into four equal parts.

We use Quantiles for in Naive Bayes where we converted continuous to categorical variables.

3.5 Problems with the data

The reason why Naive Bayes network failed is due to the data. So, every user location in the data had either 332 or 0 records. The 0 records were due to the fact these records contain missing values. Thus, that's why it is difficult to figure out whether two locations are dependent or not. Due to this, it is difficult to obtain prior probabilities for the countries also.

For example, if we return to the basics of probability theory, so, if there are two variables X and Y the idea is that X tells something about Y. So, if $X = 1$, it is almost certain that $Y = 1$ and this is expressed as $P(Y = 1|X = 1) > P(Y = 1)$. Therefore, this means that both X and Y are dependent. But, when we convert this to frequencies of occurrences and take a note of all the cases occurring for $X = 1$, there will obviously be more cases for $Y = 1$ even though it is relative to the number of cases with $X = 1$ versus $Y = 1$ and total number of cases. So, in our case, $P(Y = 1)$ is always uniform (0.5 in the binary case, 1/3 with 3 values, etc). This is due to the way the data is collected. This means that we won't have the prior probabilities for $P(Y)$ and also $P(X)$.

Due to the problem with the data and also for obtaining prior probabilities, we will reconstruct the data. This is explained in the next section.

3.6 Finding the right Bayesian network structure

Next we try to find the right Bayesian network structure where we learn the network using Tabu network.

So, the algorithm for this is as follows:

- Initially, we consider only a small subset of countries.
- Thus, after that all the regional codes are modified to country code. For example, NL1 is modified to NL.
- After that, an association list is created so that it contains information about all the regions in a country and its population.
- This association list will help us to compute the total population per country.
- The next step is to learn a Bayesian network using Tabu search and it finds the right structure. The probabilities of both user and friends location will be uniform since we used one variable to calculate both the features

3.7 Package for constructing Bayesian networks

The package which we use for constructing Bayesian networks is the bnlearn package from R. bnlearn is a Bayesian network structured learning package which is used for learning the probabilistic graphical models. We can implement structure learning algorithms like constraint based learning algorithm, score based learning algorithm, Bayesian network classifiers.

4. Results

4.1 Analysis

4.1.1 Naive Bayes

A Naive Bayes network is created for predicting the social connectedness index based on the distance and population features.

Quantiles

Initially, the variables which were continuous for every input and output feature are converted to categorical using quantiles. So, after constructing the Naive Bayes network in this case, the accuracy is found to be 30 percent. The confusion matrix can be seen in table 6.1 The confusion matrix shows that there is equal prediction for every level.

Table 4.1: Confusion matrix when using quantiles

nb.pred	0	1	2	3
0	1030	707	610	633
1	638	730	706	659
2	384	497	552	453
3	577	681	715	837

Based on these outcomes, we can conclude that Naive Bayes is not the right fit due to the problem of accuracy and that's in the next section we will try to find the right Bayesian network structure.

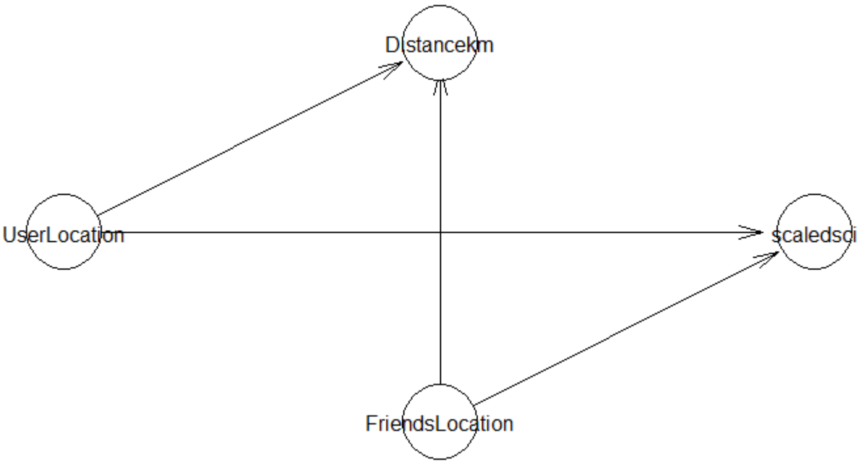


Figure 4.1: Bayesian Network plot

4.2 Trying to find the structure of Bayesian network

In all the previous experiments, all the regions in the countries were identified by their codes but to answer the second research question, we will remove all the codes and consider every country as one. After that, we will learn a Bayesian network using Tabu based search.

4.2.1 Bayesian network

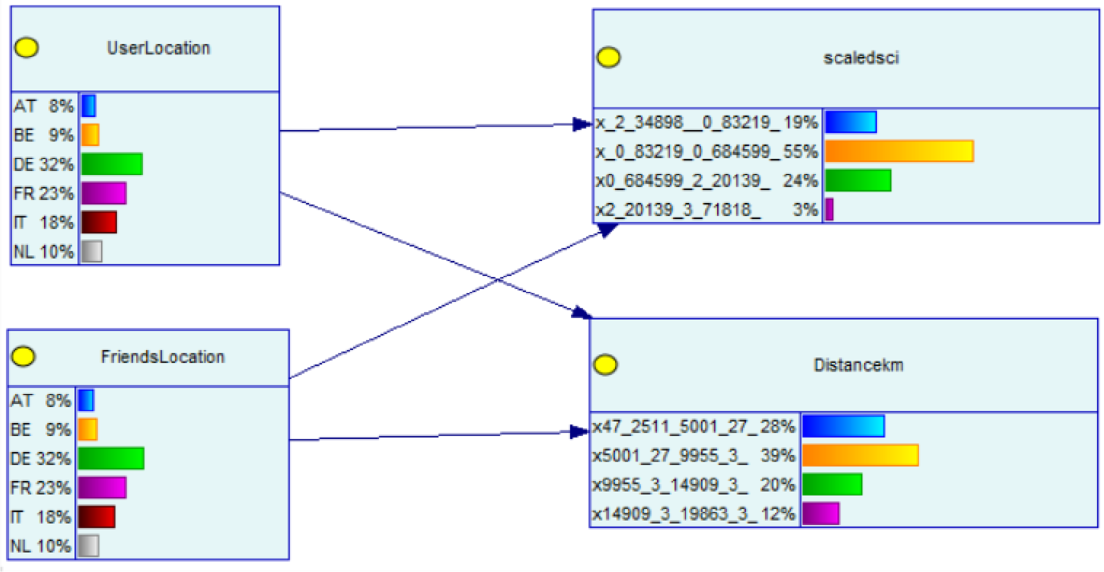


Figure 4.2: Bayesian Network

The plot for the Bayesian network is shown in 4.1 and the Bayesian network is seen in 4.2. So, as seen in the figure, there are four nodes: UserLocation, FriendsLocation and scaledsci. We also use only a subset of the data (selecting some regions). The probabilities obtained are learnt from the data. So, we use the population feature to obtain the prior probabilities for the nodes UserLocation and FriendsLocation and that's why the prior probabilities for both the nodes (UserLocation and FriendsLocation) are similar. We discretize the distance and scaledsci features by using intervals of discretization. After this, we construct a Bayesian network using Tabu based search.

4.2.2 Prior probabilities obtained for countries

The prior probabilities are calculated as follows: For example, if we have 5 countries, we will obtain the total population per country separately. We will then calculate the overall population of all the 5 countries. So, now, to obtain the prior probabilities we will divide every country's individual population to the overall population of all the countries. This will give us the prior probability.

Table 4.2: Prior probabilities

Country name	Prior probability
Austria	0.047
Belgium	0.061
France	0.360
Germany	0.439
Netherlands	0.092

The prior probabilities obtained for countries are given in table 6.2. As expected, the prior probability of Germany is more since the population in Germany is high and also the size of the country is larger relatively as opposed to other countries. The lowest prior probability is obtained for Austria since the population in Austria is relatively small and the size of the country is also smaller.

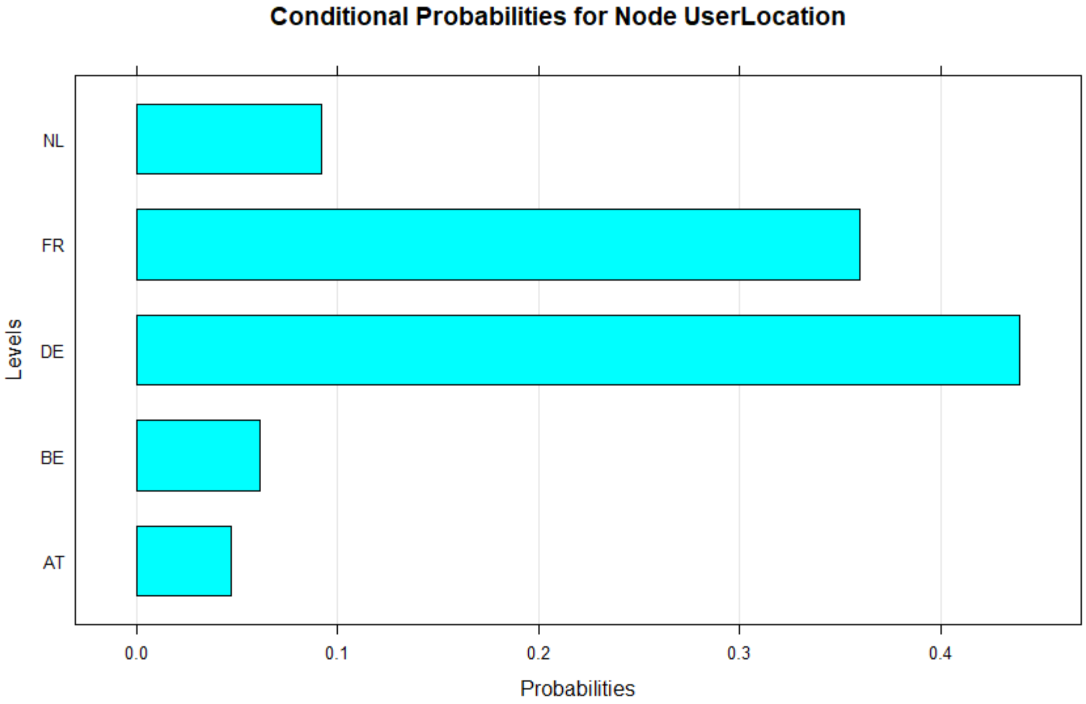


Figure 4.3: CPT plot

This can be verified by looking into the Conditional probability plot in 4.3. As expected, the probability level of Germany (DE) and France (FR) is high when we compare it to other countries.

4.2.3 Validating the network

For cross validation, we use 10 folds and the structure learning algorithm which we use is the 'hill climbing' and we evaluate the network using 2 score functions namely : Bayesian Information Criterion (BIC) and Bayesian Likelihood Equivalence (BDe). The plot for the same is shown in the 4.4. The Log-Likelihood loss for both BIC and BDe is found to be 5.19. The comparison between both the score function is almost similar.

According to [22], if a score is below 6, it is given that the model is good. So, we can conclude that since we obtained for a score of 5.19, we can safely conclude that the model has performed well.

4.2.4 Evaluating the model

We evaluate the model using these basic evaluation methods: ROC and confusion matrix. The Receiver Operating Characteristic (ROC) curve can be seen in 4.5. We have obtained 76.5 of AUC. In the figure, since the curve is closer to the top left corner, the model has performed well.

The confusion matrix plot can bee seen in 4.6. In the figure, the accuracy of the model is found to 88 percentage and as seen in the figure, there are four classes and all the data points are distributed for the four classes.

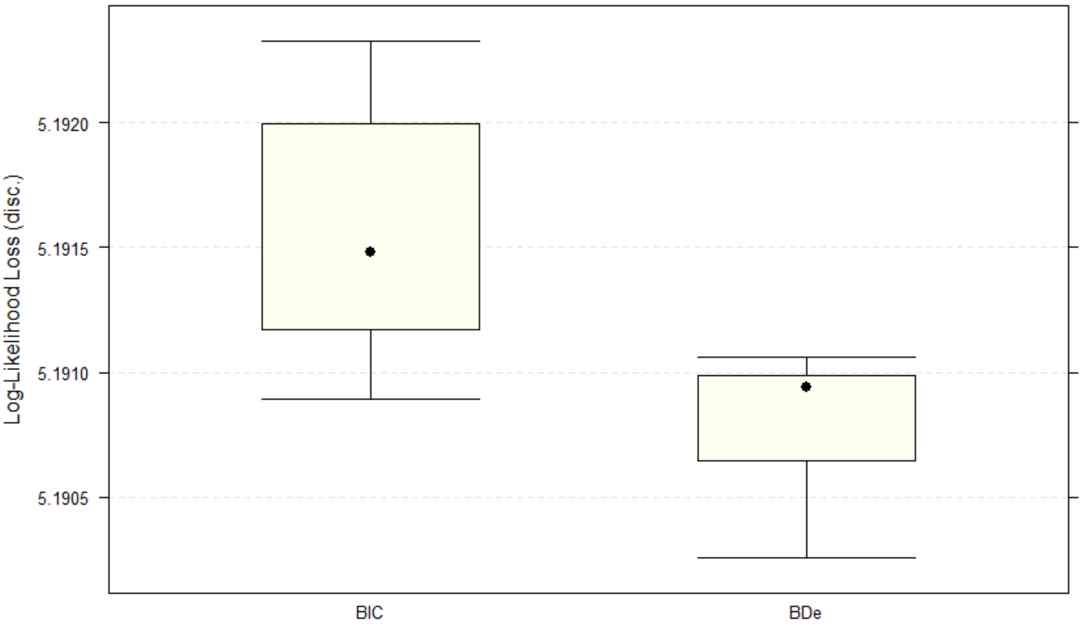


Figure 4.4: K Cross validation

4.2.5 Brier score

The Brier score of the model is found to be 0.1164788. According to [23], a brier score of 0 means perfect accuracy and a score of 1 means imperfect accuracy. Therefore, since the score which we obtained is so close to 0, this means that our model performed well.

4.2.6 Comparison to the Naive Bayes model

When we compare the current model to the Naive Bayes model, the Naive Bayes model has obtained an accuracy of 30 percent and also, there is a problem of data imbalance. The problem of data imbalance however didn't occur for the current model. There is also a massive difference in accuracy since the current model has achieved 87 percentage of accuracy.

The log-likelihood of the current model is -189082.8 and for Naive Bayes, it is -205864.2. According to [24], the higher the likelihood of the model, better is the fit of the model. Thus, as proved already by the accuracy and also by the problem of data imbalance, this is also an other parameter which proves that the current model has fit very well than the Naive Bayes.

4.2.7 Calibration plots

For obtaining the calibration plots, initially we will use the Graphical Independence Network package or gRain. We will pass the fitted Bayesian network to it. After this, using query grain, we will obtain the marginal probabilities for the required feature. This shall give us the marginal probabilities for the predicted feature and for obtaining the actual marginals, we will divide the sum of individual class of every feature to the total sum of all the classes for that feature.

The calibration plots for the features distancekm, scaledsci are shown from 4.7, and 4.8 respectively. In these plots, we compare the marginals of every feature with with the actual probabilities. From the figures 4.7, 4.8 , the calibration of both distancekm and scaledsci is good.

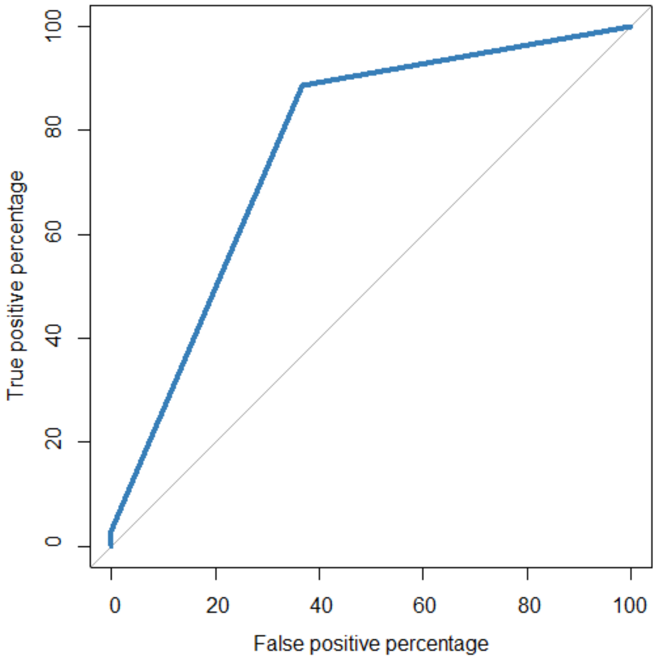


Figure 4.5: ROC curve

4.2.8 Information obtained from the model

The distribution of SCI and distance is shown in figures 4.9 - 4.11 by using two different countries as user location and friends location every time.

As it is clearly visible from the figures, the distribution varies for all the different combination of countries.

4.2.9 Number of bins

The confusion matrix plots obtained for bins 2, 3, 5, 6 are shown from the figures 4.12 - 4.15 respectively. The accuracy decreases for every bin and also there is a problem of data imbalance for bins 3, 5, 6.

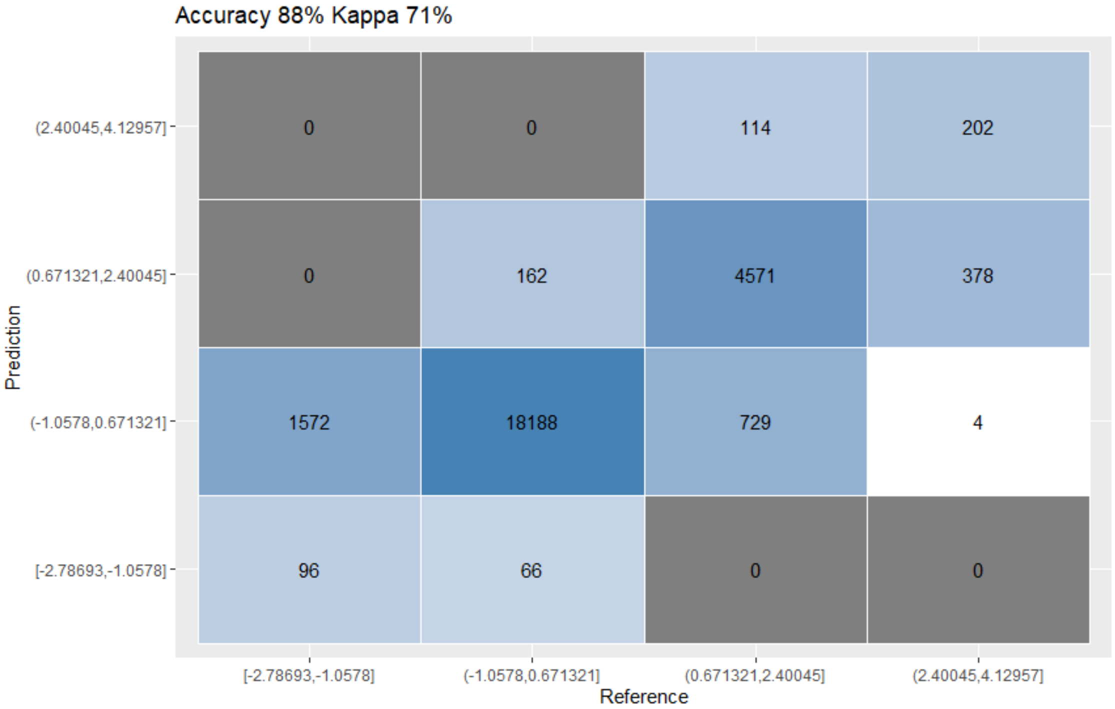


Figure 4.6: Confusion matrix plot (4 bins)

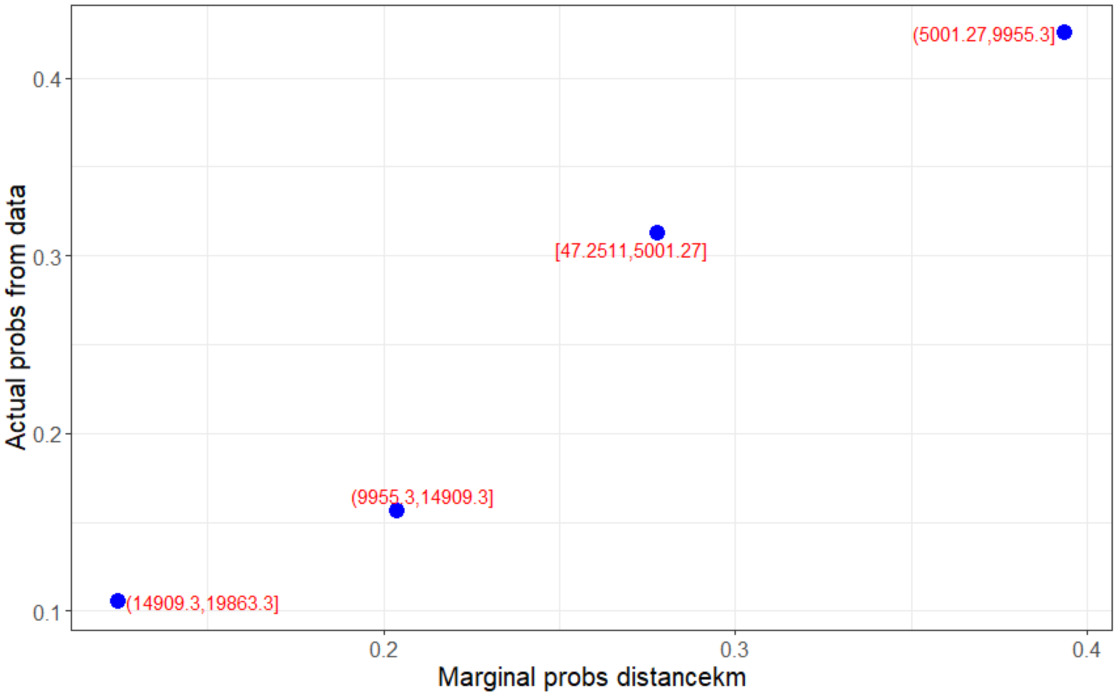


Figure 4.7: Marginal probability plot for distancekm

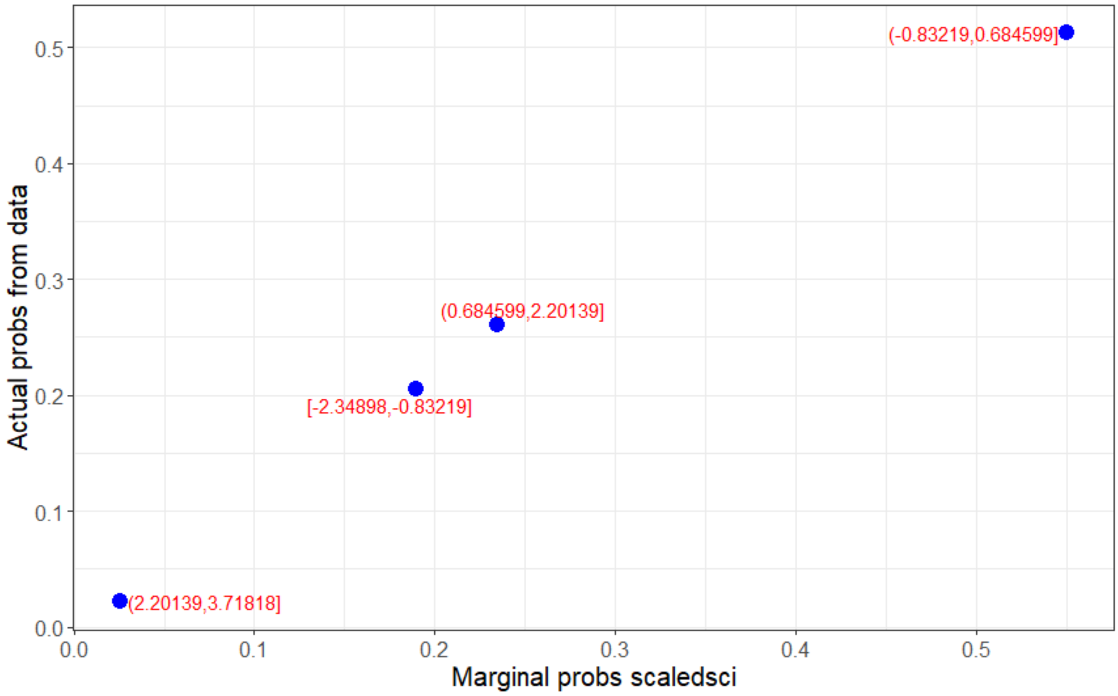


Figure 4.8: Marginal probability plot for scaledsci

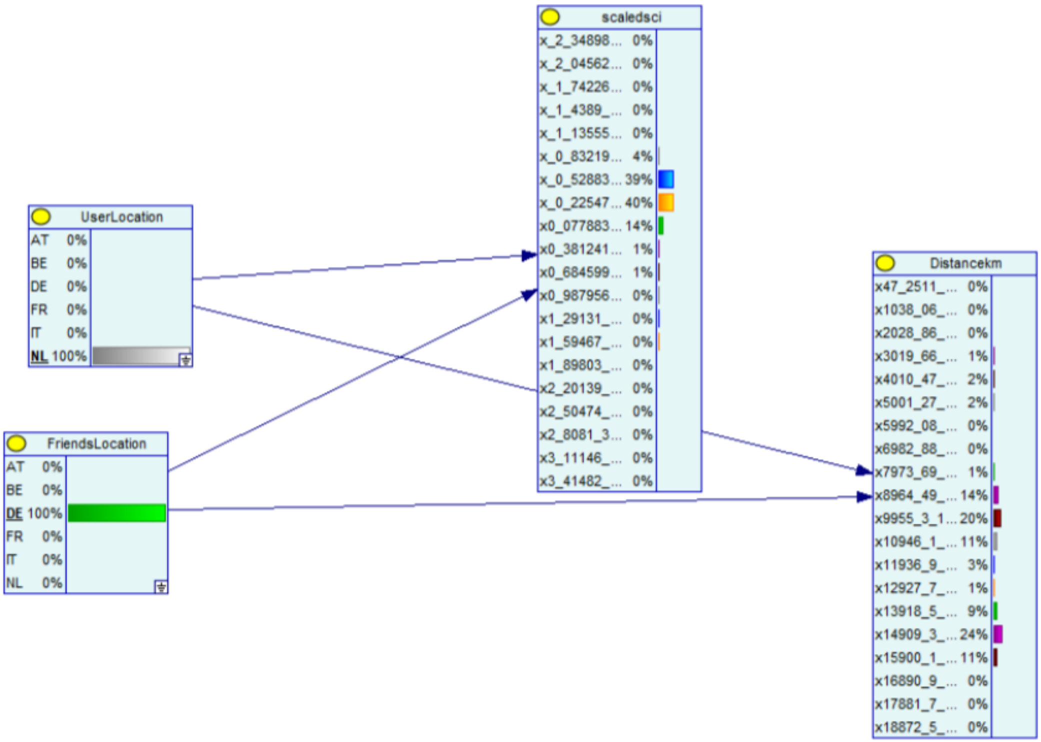


Figure 4.9: Distribution of distance and scaledsci (Netherlands and Germany)

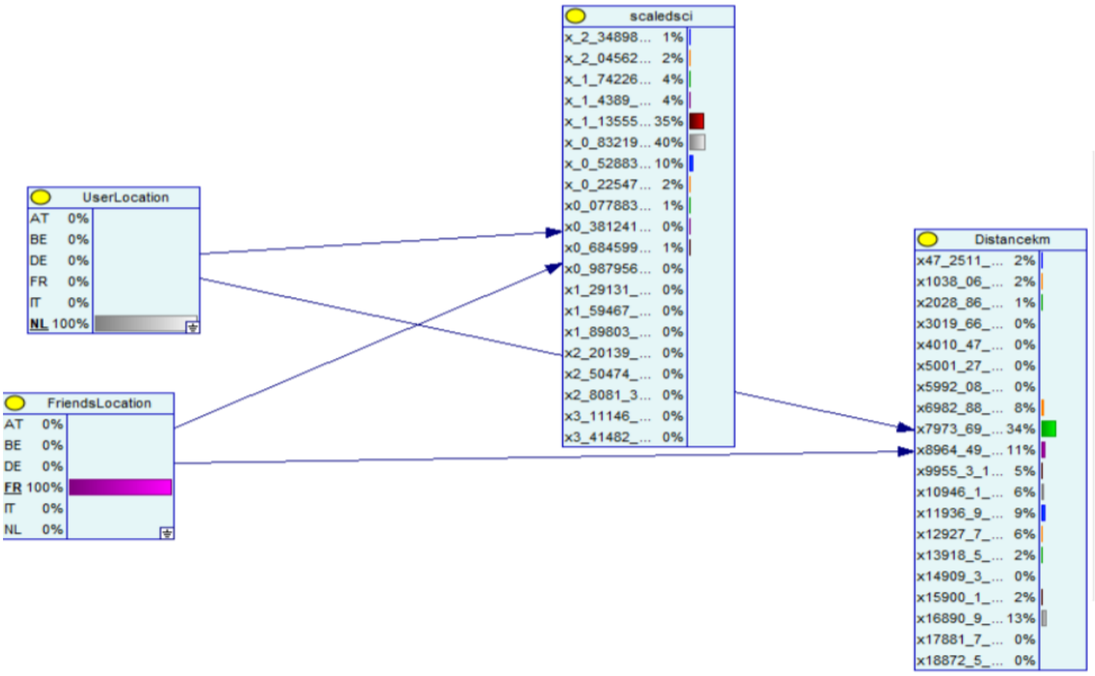


Figure 4.10: Distribution of distance and scaledsci (Netherlands and France)

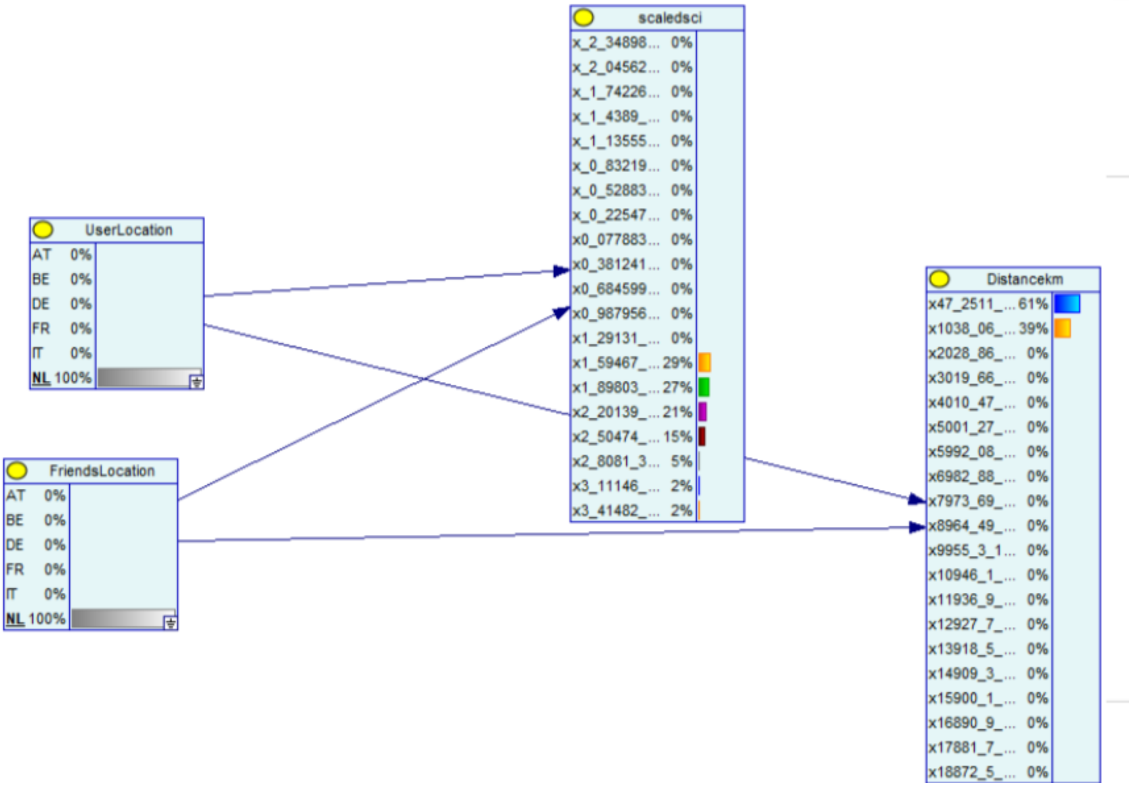


Figure 4.11: Distribution of distance and scaledsci (Netherlands and Netherlands)

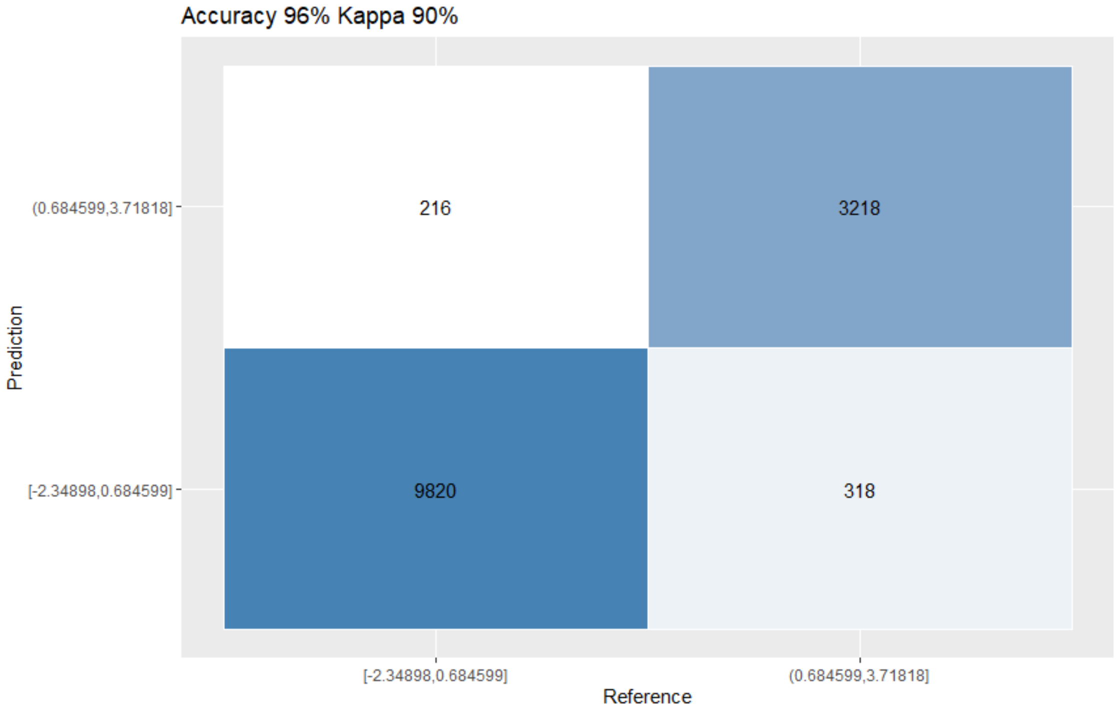


Figure 4.12: Confusion matrix plot (2 bins)

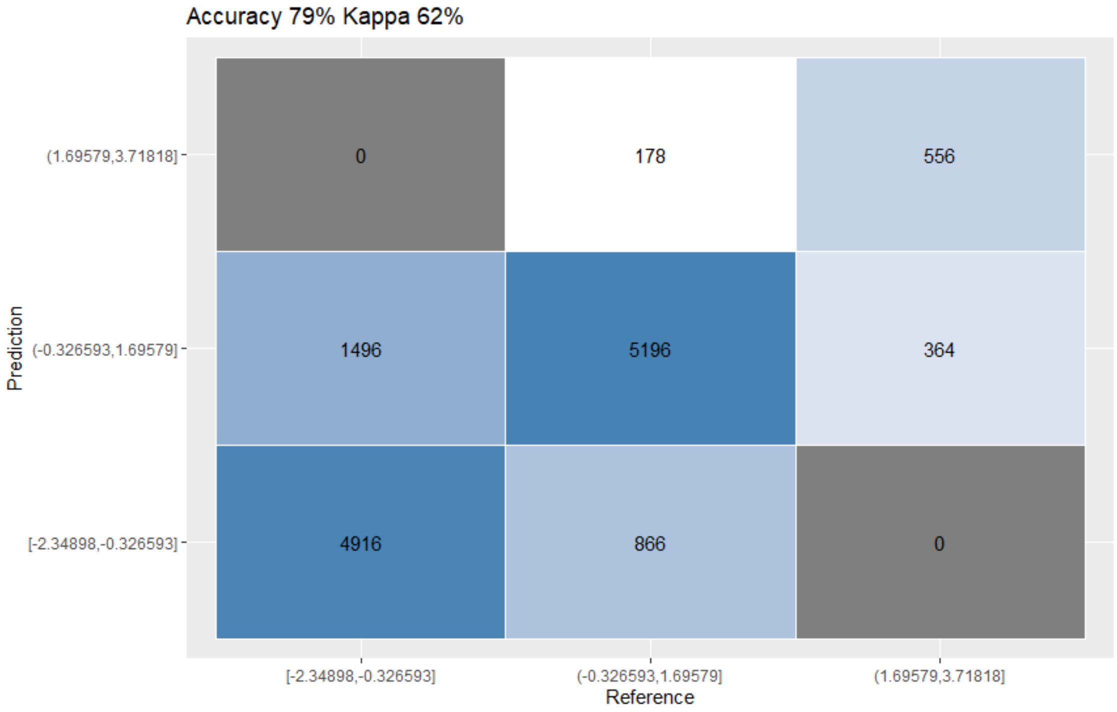
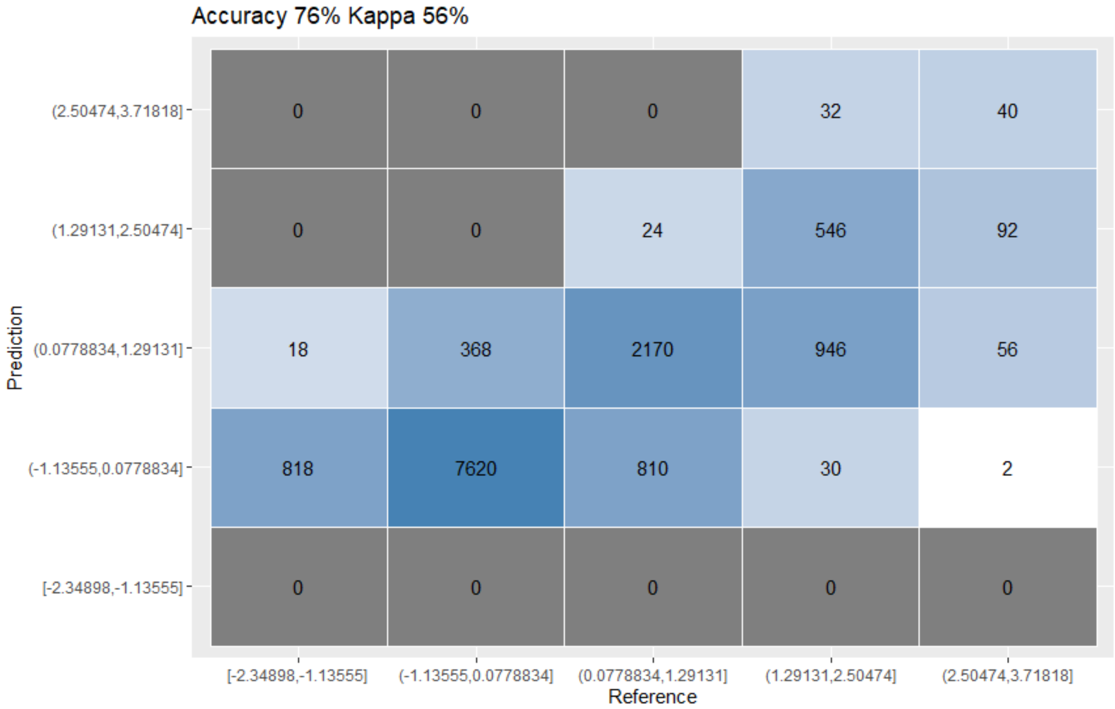


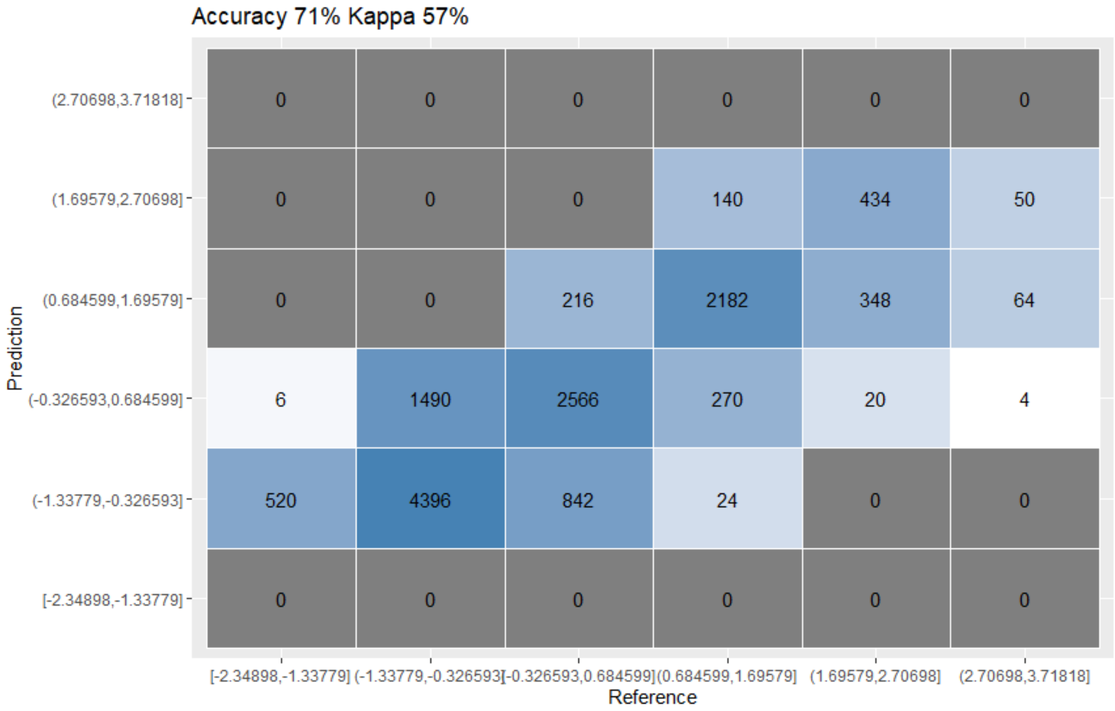
Figure 4.13: Confusion matrix plot (3 bins)



Accuracy 76% Kappa 56%

Prediction \ Reference	[-2.34898,-1.13555]	(-1.13555,0.0778834]	(0.0778834,1.29131]	(1.29131,2.50474]	(2.50474,3.71818]
(2.50474,3.71818]	0	0	0	32	40
(1.29131,2.50474]	0	0	24	546	92
(0.0778834,1.29131]	18	368	2170	946	56
(-1.13555,0.0778834]	818	7620	810	30	2
[-2.34898,-1.13555]	0	0	0	0	0

Figure 4.14: Confusion matrix plot (5 bins)



Accuracy 71% Kappa 57%

Prediction \ Reference	[-2.34898,-1.33779]	(-1.33779,-0.326593]	(-0.326593,0.684599]	(0.684599,1.69579]	(1.69579,2.70698]	(2.70698,3.71818]
(2.70698,3.71818]	0	0	0	0	0	0
(1.69579,2.70698]	0	0	0	140	434	50
(0.684599,1.69579]	0	0	216	2182	348	64
(-0.326593,0.684599]	6	1490	2566	270	20	4
(-1.33779,-0.326593]	520	4396	842	24	0	0
[-2.34898,-1.33779]	0	0	0	0	0	0

Figure 4.15: Confusion matrix plot (6 bins)

4.3 Discussion

4.3.1 Problems with Naive Bayes network and Tree Augmented Naive Bayes (TAN)

Initially, before going with an Search based network, we constructed both Naive Bayes and TAN. But, we encountered problems due to Naive Bayes and TAN. Initially, We converted the features which were continuous to discrete using Quantiles, manual conversion, intervals of discretization. But, the problems which were faced for both Naive Bayes and TAN are low accuracy, class imbalance. The class imbalance didn't occur with quantiles but the accuracy was on the lower side. So, when we tried converting from continuous to discrete manual, we faced data imbalancing problems (unequal distribution of datapoints). The same problem was faced when we used intervals of discretization.

So, in order to overcome both of these problems, we tried reconstructing the social connectedness index from the data itself. The accuracy which we obtained for both Naive Bayes and TAN were pretty good but again the problem of data imbalancing occured and there was a massive gap between the intervals in the confusion matrix. The next thing we tried was reducing the domain size of the data and we tried converting the continuous to categorical using the same methods described above. But the same problem of data imbalance occurred. Furthermore, we also tried a different way of reducing the domain size of the data i.e. by removing the outliers in the data and removing the rows which had NA/NAN values. Even then, the same problem occured all over again.

4.3.2 A search based network

To resolve the problems which we faced due to Naive Bayes and TAN, we constructed a search based network. In particular, the way the pre-processing was done was different. We used one variable (population feature) to calculate the prior probabilities for both user location and friends location. This is the reason why the prior probabilities of both user location and friends location are similar. Moreover, we also considered every country as one. We also validated, evaluated the network. The model now didn't have the problems of low accuracy or class imbalancing.

Overall, the network performed very well in terms of accuracy and also the calibration plots which were obtained for scaledsci and distance km proved that. The brier score obtained also was close to 0 which again proved that the model performed well. The scores which we obtained for BDe and BIC for K-Cross validation were also pretty good which again proved that the model performed. Again in the ROC curve obtained, many curves were close to the top left corner which proves again that the model performed very well.

We also saw if any useful information can be obtained from the model. Therefore, we compared the network by using three different combination of countries and to see the distribution of scaled sci and distancekm. We compared the network for the country combinations : (Netherlands, Germany), (Netherlands, France) and (Netherlands, Netherlands). We saw that the distribution was different for every combination of countries and it wasn't similar.

4.3.3 Comparison with the Naive Bayes network

We also compared the search based network with the Naive Bayes network. The same problem of data imbalance occurred for Naive Bayes and also the accuracy of the model was on the lower side. We also compared the model in terms of Log-Likelihood and the search based model performed well in terms of Log-Likelihood too.

4.3.4 Number of bins

The number of bins which we used for the main network was 4 and the reason for using 4 bins is because when 4 bins were used, the model didn't face problem of accuracy and data imbalance which was faced for other bins that is reflected also on the confusion matrix plots.

Furthermore, when we tried to obtain some information from the network i.e.to see the distribution of scaledsci and distance for different sets of countries, this is not possible when we use less bins. Thus, that's why when we try to obtain information from the model, we used 20 bins and that's why we were able to see the distribution for all the combination of countries. This can be verified by seeing the 4.17, 4.18 and 4.16. It is clearly visible in these figures that the distributions are not possible to be seen using 12 bins, 14 bins and 17 bins but only possible when we use bins higher than this. But, since the number of bins are increased to see the distribution of scaledsci and distancekm, we needed to compromise on the accuracy. When we obtained the distributions for scaledsci and distancekm when using 20bins, the accuracy we obtained was 40 percentage.

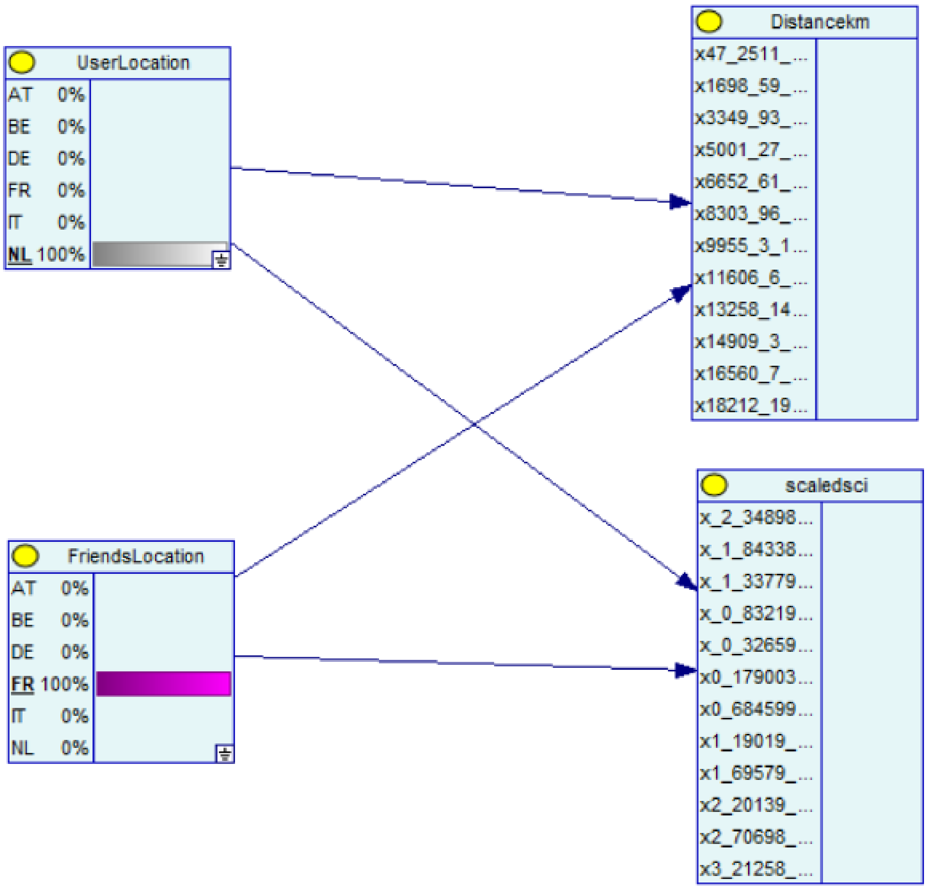


Figure 4.16: Network comparison for 12 bins

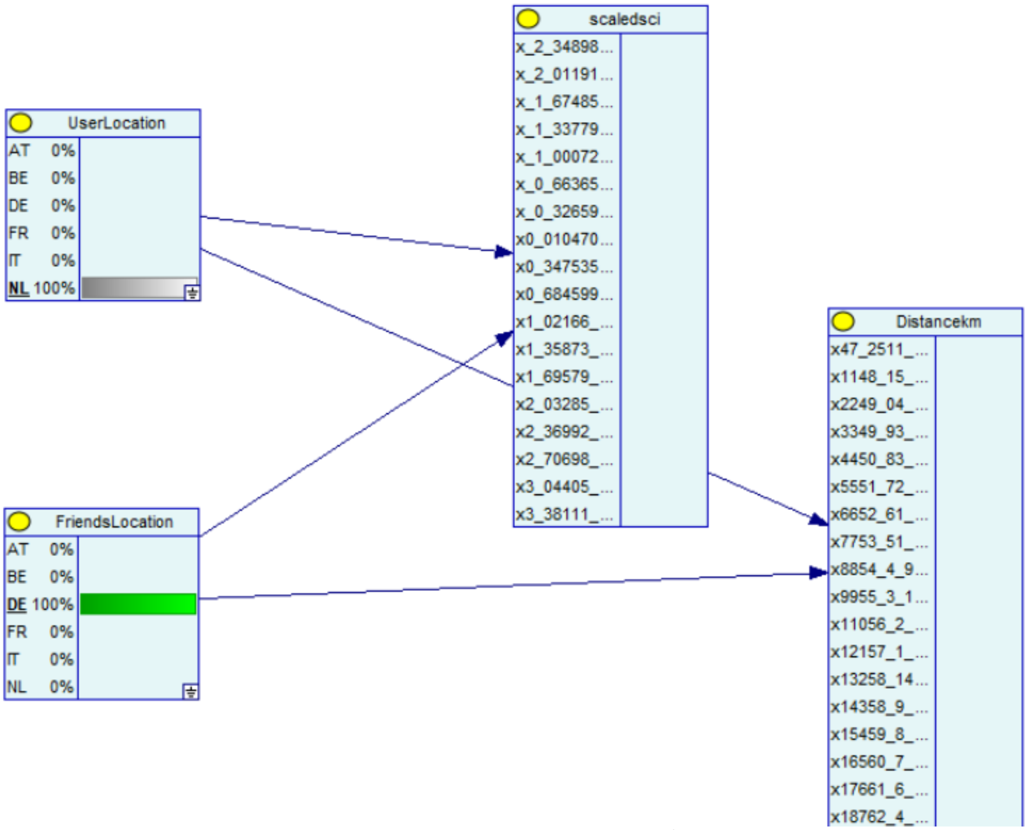


Figure 4.17: Network comparison for 17 bins

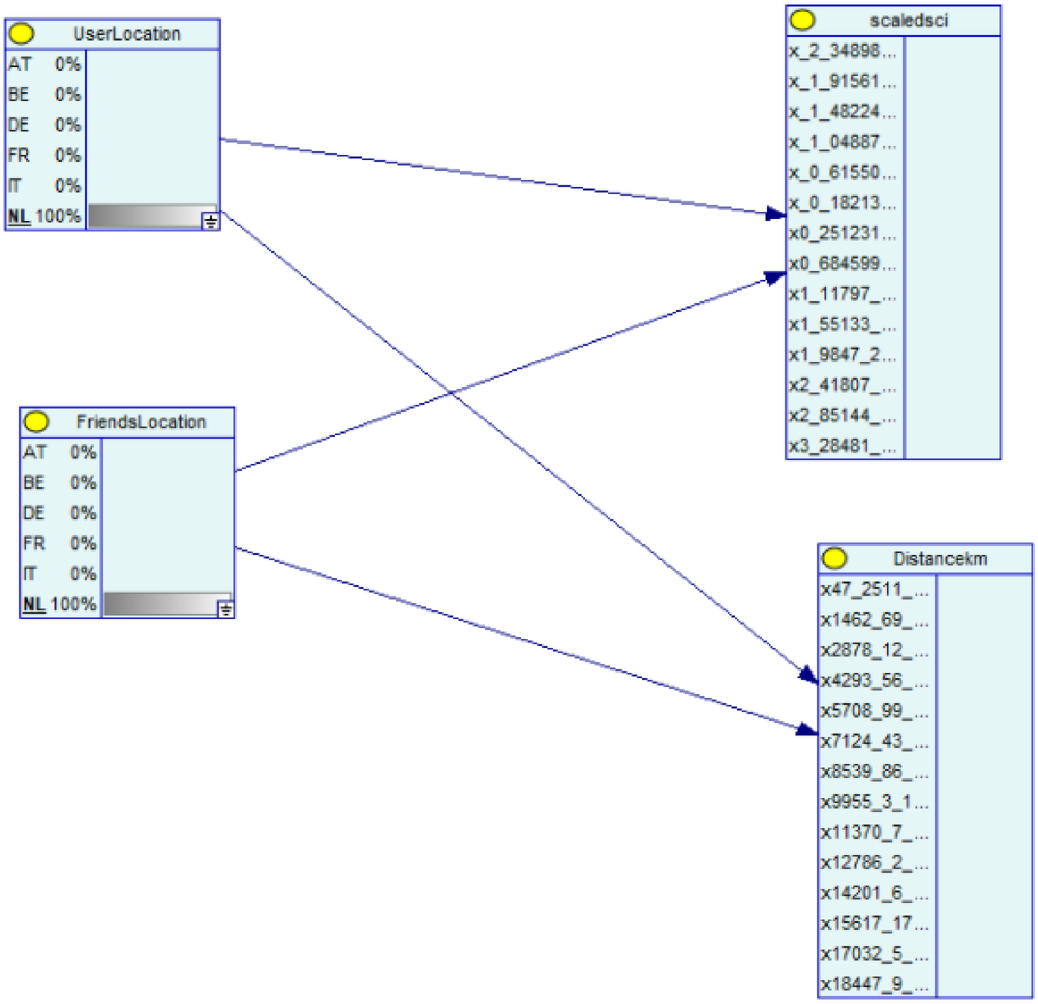


Figure 4.18: Network comparison for 14 bins

4.4 Future recommendations

We have constructed a search based network based on structure learning. To improve the network, we could maybe add more features and see how the outcome is. But, the research conclusions indicate that it it is a good idea to move forward with Bayesian networks for this problem since even for obtaining a normal search based network, it took a long time to decipher what kind of information is needed. We encountered a lot of problems during the start of research when we tried to construct a TAN and a Naive Bayes network. We also tried solving this problems using many methods and none of them worked. Furthermore, we somehow figured out a way to construct a network and also obtained some useful information from the network. But, the results from the research show that constructing a Bayesian network will be so hard for this problem and it is not a good idea to move further with Bayesian networks.

5. Conclusion

In this thesis, we try to find the right Bayesian network structure for predicting the Social Connectedness Index (SCI). SCI indicates the strength between two countries and we tried to learn a Bayesian network so that it will learn the parameters of the data. So, to achieve this, initially we constructed a Naive Bayes network and found that it performed poorly in terms of accuracy and also it encountered the problem of data imbalance. Thus, after this, we tried to learn the Bayesian network using tabu search and it helped us in finding the right structure of the network. Therefore, to achieve this, we used the population feature from the data to calculate the prior probabilities of user location and friends location.

We also validated the network using K-cross validation and by using calibration plots. The scores which we obtained for K-cross validation showed that the model performed well and the calibration plots for distancekm, scaledsci implied that the calibration of the model was good. We also evaluated the network using confusion matrix and ROC curves. From the confusion matrix, we found that the model achieved an accuracy of 87 percentage and there was no problem of data imbalancing. From the ROC curve, we found that the curve is close to the top left corner which gave a further indication that the model performed well. We also calculated the Brier score of the model and found that it was close to 0 which again proved that the model performed well.

After this, we compared the model to the Naive Bayes network and found that Naive Bayes performed poorly in terms of accuracy, Log-Likelihood. The accuracy which we obtained for Naive Bayes was 67 percentage and the Log-Likelihood score was also higher. We also compared the network for various types of bins and found that when the number of bins increases, the accuracy decreases and also the problem of data imbalance occurs. We also try to obtain some useful information from the network when we compared the network for different combination of countries. This was only possible when the number of bins were 20 and more than that. We found that the distribution of scaledsci and distancekm varied for every combination of countries. But, since the number of bins were increased, we needed to compromise on the aspect of accuracy.

Bibliography

[1] Maeve Duggan, Nicole B Ellison, Cliff Lampe, Amanda Lenhart, and Mary Madden. Social media update 2014. *Pew research center*, 19:1–2, 2015. 1

[2] David Koelle, Jonathan Pfautz, Michael Farry, Zach Cox, Geoffrey Catto, and Joseph Campolongo. Applications of bayesian belief networks in social network analysis. In *Proceedings of the 4th Bayesian modeling applications workshop during the 22nd annual conference on uncertainty in artificial intelligence*, 2006. 2

[3] Damien R Farine and Ariana Strandburg-Peshkin. Estimating uncertainty and reliability of social network data using bayesian inference. *Royal Society open science*, 2(9):150367, 2015. 3

[4] S. H. Shalforoushan and M. Jalali. Link prediction in social networks using bayesian networks. In *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pages 246–250, 2015. 3

[5] Michael Fire, Lena Tenenboim, Ofrit Lesser, Rami Puzis, Lior Rokach, and Yuval Elovici. Link prediction in social networks using computationally efficient topological features. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pages 73–80. IEEE, 2011. 3

[6] Nesserine Benchettara, Rushed Kanawati, and Celine Rouveirol. Supervised machine learning applied to link prediction in bipartite social networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 326–330. IEEE, 2010. 3

[7] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007. 4

[8] Elaheh Nasiri, Asgarali Bouyer, and Esmaeil Nourani. A node representation learning approach for link prediction in social networks using game theory and k-core decomposition. *The European Physical Journal B*, 92(10):1–13, 2019. 4

[9] Yang Zhang and Jun Pang. Distance and friendship: A distance-based model for link prediction in social networks. In *Asia-Pacific Web Conference*, pages 55–66. Springer, 2015. 4

[10] Dimitris Margaritis. Learning bayesian network model structure from data. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2003. 4

[11] Agnieszka Oniśko, Marek J Druzdzel, and Hanna Wasyluk. Learning bayesian network parameters from small data sets: Application of noisy-or gates. *International Journal of Approximate Reasoning*, 27(2):165–182, 2001. 4

[12] Ira Cohen, Nicu Sebe, FG Gozman, Marcelo Cesar Cirelo, and Thomas S Huang. Learning bayesian network classifiers for facial expression recognition both labeled and unlabeled data. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003. 4

[13] Nir Friedman, Iftach Nachman, and Dana Pe'er. Learning bayesian network structure from massive datasets: The" sparse candidate" algorithm. *arXiv preprint arXiv:1301.6696*, 2013. 5

[14] Tommi Jaakkola, David Sontag, Amir Globerson, and Marina Meila. Learning bayesian network structure using lp relaxations. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 358–365. JMLR Workshop and Conference Proceedings, 2010. 5

[15] Pedro Larranaga, Mikel Poza, Yosu Yurramendi, Roberto H. Murga, and Cindy M. H. Kuijpers. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE transactions on pattern analysis and machine intelligence*, 18(9):912–926, 1996. 5

[16] Cassio P De Campos, Zhi Zeng, and Qiang Ji. Structure learning of bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 113–120, 2009. 5

[17] Lobna Bouchaala, Afif Masmoudi, Faiez Gargouri, and Ahmed Rebai. Improving algorithms for structure learning in bayesian networks using a new implicit score. *Expert Systems with Applications*, 37(7):5470–5475, 2010. 5

[18] Mikko Koivisto and Kismat Sood. Exact bayesian structure discovery in bayesian networks. *The Journal of Machine Learning Research*, 5:549–573, 2004. 6

[19] Michael Bailey, Ruiqing Rachel Cao, Theresa Kuchler, Johannes Stroebel, and Arlene Wong. Measuring social connectedness. Technical report, National Bureau of Economic Research, 2017. 7, 8

[20] Michael Bailey, Theresa Kuchler, Dominic Russel, Johannes Stroebel, et al. Social connectedness in europe. 2020. 8

[21] Niels Radstake, Peter JF Lucas, and Elena Marchiori. Learning bayesian models using mammographic features. 2010. 14

[22] Jimmy Wales. https://en.wikipedia.org/wiki/bayesian˙information˙criterion, February 2020. 28

[23] University of Virginia Library. https://data.library.virginia.edu/a-brief-on-brier-scores/, May 2018. 29

[24] Analyttica datalab. https://medium.com/@analyttica/log-likelihood-analyttica-function-series-cb059e0d379, February 2019. 29

Acknowledgment

First and foremost, I would like to thank my parents who always believed and loved me even though the going got tough and they encouraged me in every aspect of my life. Without them, I won't be the person I'm today. Without them, I wouldn't have achieved even a single part of this. I owe this to them. Even when I was in a tough mental spot, the only thing which got me going was my parents.

I wish to express my sincere appreciation to my supervisor, Professor Peter Lucas, who has the substance of a genius: he convincingly guided and encouraged me to be a professional and do the right thing even when the road got tough. Without his persistent help, the goal of this project would not have been realized. He helped me in every situation and even when I wasn't not good mentally, he encouraged me and helped me get through it. Bayesian networks were completely new to me but with his persistent help, guidance and support of Peter it was not that difficult and it helped not only gain knowledge but also helped me complete my thesis

I also would like to thank my other supervisor, Professor Clara Stegehuis, for giving me the opportunity to work under her for this project. Her advises and insights were unmatchable. Since I'm from a Computer science background, it was difficult at first since this topic was mathematical. But due to Clara's guidance and enthusiasm, it helped me to learn the mathematical side of things. Her prompt inspirations, timely suggestions, enthusiasm and dynamism have made me complete my thesis.

I also would like to thank my friends, relatives for their constant love and support. I would like give a special mention to my brothers, Ashwin and Lokesh for their ever lasting support and appreciation. I would also like to give a special mention to my friends, Charan, Shyam, Prince, Bharath Kumar and Naga who always listened to my problems and always were there as my support system.