THE UNIVERSITY OF
SYDNEY

INFERRING ANOMALIES FROM DATA USING BAYESIAN NETWORKS

A thesis submitted in fulfilment of the requirements for the
degree of Doctor of Philosophy in the School of Information Technologies at
The University of Sydney

Sakshi Babbar
August 2013

I have examined this thesis and attest that it is in a form suitable for examination for the degree of Doctor of Philosophy.

_____

(Professor Sanjay Chawla) Principal Adviser

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

_____

Sakshi Babbar

Sydney
August 27, 2013

This work was carried out under the supervision
of Professor Sanjay Chawla

Acknowledgements

I am greatly indebted to many people who have contributed to this thesis. My sincerest gratitude goes to my supervisor, Professor Sanjay Chawla, who supported me throughout my study with his patience and knowledge. I appreciate all his contributions of time, ideas, and financial support that made my Ph.D. experience productive and stimulating. It was an extreme honor to work under you Professor. I also take this as an opportunity to thank my associate supervisor, Dr. Taso Viglas, for his friendly and supportive gesture during my entire study period.

I thank my cherished friends for providing enthusiasm and empathy throughout my stay on a foreign land. The wonderful companionship of Meena, Linsey, Nadia, Tara, Khoa, Wei, Didi and Kalpana made my study years enjoyable. My thanks goes to all of them. Unusual though, I also deeply acknowledge the pleasing company given to me by the online music radio stations. Listening to them was always rejuvenating.

Last but not least, I thank my family for their unconditional love and encouragement so that I could live up to my dream of studying a doctorate course. My special thanks goes to my father who has been a continuous source of support and positivity; thank you daddy.

Publications

1. A Causal Approach For Mining Interesting Anomalies
   Sakshi Babbar, Didi Surian and Sanjay Chawla
   In proceedings of the 26th Canadian Conference on Artificial Intelligence,
   Regina, Canada, 2013, pp. 226-232.

2. Mining Causal Outliers using Gaussian Bayesian Networks
   Sakshi Babbar and Sanjay Chawla
   In proceedings of the 24th IEEE International Conference on Tools
   with Artificial Intelligence,
   Athens, Greece, 2012, pp. 97–104.

3. On Bayesian Network and Anomaly Detection
   Sakshi Babbar and Sanjay Chawla
   In proceedings of the 16th International Conference on Management of Data,
   Nagpur, India, 2010, pp. 125-137.

4. Integration of Domain Knowledge for Outlier Detection in High Dimensional Space
   Sakshi Babbar and Sanjay Chawla
   In proceedings of the 14th International Conference on Database Systems
   for Advanced Applications Workshop,
   Brisbane, Australia, 2009, pp. 363–368.

Abstract

Existing studies on data mining has largely focused on the design of measures and algorithms to identify outliers in large and high dimensional categorical and numeric databases. However, not much stress has been given on the interestingness of the reported outlier. One way to ascertain interestingness and usefulness of the reported outlier is by making use of domain knowledge. In this thesis, we present measures to discover outliers based on background knowledge, represented by a Bayesian network. Using causal relationships between attributes encoded in the Bayesian framework, we demonstrate that meaningful outliers, i.e., outliers which encode important or new information are those which violate causal relationships encoded in the model. Depending upon nature of data, several approaches are proposed to identify and explain anomalies using Bayesian knowledge.

- We propose a novel approach which combines the use of Bayesian network and probabilistic association rules to discover and explain anomalies in categorical data set. The Bayesian network allows us to organize information in order to capture both correlation and causality in the feature space, while the probabilistic association rules have a structure similar to association mining rules. In particular, we focus on two types of rules: (i) *low support & high confidence* and, (ii) *high support & low confidence*. New data points which satisfy either one of the two rules conditioned on the Bayesian network are the candidate anomalies.

- We design a measure to discover outliers in numerical data sets and data sets containing mixture of data types using the domain knowledge captured by a Gaussian Bayesian network and Hybrid Bayesian network respectively. By first constructing a Bayesian network, depending upon type of data set, we identify those data points as outliers which violate casual relationships encoded in the model.

- Outliers are often identified as data points which are “rare”, ”isolated”, or ”far away from their nearest neighbors”. We show that these characteristics may not be an accurate way of describing interesting outliers. Through a critical analysis on several existing outlier detection techniques, we show why there is a mismatch between outliers as entities described by these characteristics and “real” outliers as identified using Bayesian approach.

- Measures that we propose in this thesis are specially designed to give contextual information of an anomaly, i.e., our approaches provide an explanation for the outliers discovered. This in turn can be used to enrich our knowledge about the underlying data generating process.

- We show that the Bayesian approaches presented in this thesis has better accuracy in mining genuine outliers while, keeping a low false positive rate as compared to traditional outlier detection techniques.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Defining Anomalies

An anomaly is an observation in data acquired from a domain that is markedly "different" from other observations. These different observations are also referred as *outliers*, *surprises*, *unusual*, *rare* or *exceptions*. Hawkins [37] defined "an outlier is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism".

Anomaly detection methods have a wide variety of applications including fraud detection for credit cards [28], public health monitoring [75] and intrusion detection for cyber-security [10]. For example, the role of anomaly detection in credit card monitoring is to detect exceptional transactions which might indicate that credit card has been stolen and is being misused. In public health monitoring, anomaly detection techniques can be used for detecting new disease outbreaks as early as possible. Intrusion detection refers to detection of malicious activity in computer systems which may be indicative of an authorized access into the system to carry out information theft or to disrupt the network.

The real cause of outlier occurrence is often unknown to users or analysts. Sometimes, the outlier can be a flawed value resulting from poor quality of the data set. In this case, no useful contextual information is conveyed by the outlier value. However, it is also possible that an outlier represents correct, though exceptional information for example anomaly detection applications listed above. In this case, the understanding of the outlier will potentially provide new information. The identification of outliers is

important both for improving the quality of original data and for providing additional unknown knowledge from the data. However, while in the first case the correct action is to remove the outlier, in the second case a proper analysis is required so as to understand why such anomalies appear in the domain and what are its causes.

There exist several factors which makes the task of anomaly detection very challenging. First, defining a region representing normal behavior and declare an observation in the data which does not belong to this normal region, as an outlier. However this simple approach is rather challenging since defining a normal region which encompasses every possible normal behavior is a non-trivial task. Second, the difficulty of obtaining enough labeled data to characterize anomalies. Hence, in most cases we need to operate in an *unsupervised setting* where only the normal behavior can be modeled and is used to discover deviations. Third, definition of normality and anomalies are typically domain specific. So a technique designed for one application may not work in other problem areas. Lastly, it is difficult to judge about the quality of the reported outlier discovered by any outlier detection technique. For example, the determination of whether an anomaly is "noise" in the data or embodies new information is challenging problem. The thesis will propose the use the Bayesian Networks to address this challenge.

Existing studies on data mining has largely focused on the design of measures and algorithms to identify outliers in large and high dimensional categorical and numeric databases. However, not much stress has been given on the interestingness of the reported outlier. Consider a hypothetical data set belonging to a certain region of the country, highlighting persons income and their expenditures. The sample data in Figure 1.1 represents relationship between persons income (X-axis) and expenditure (Y-axis). As observed, data points are roughly clustered. We name them as C_1, C_2, C_3 , C_4, C_5 and C_6 respectively. Cluster C_1 and C_3, indicates that in a given region, persons expenditure is bounded within their income. Unlike clusters C_1 and C_3, data points forming clusters C_5, C_6 indicates that there are very small percentage of people the expenditure of whom are higher than that of their income. Likewise, there are few people in region earning high but choose to spend low as represented by the cluster C_2. Lastly, a cluster C_4 indicates percentage of people with high earning and high expenditure. We ask data points that should be identified as anomalies? Whether it should be observations from clusters C_2, C_4, C_5, C_6 , data points from clusters C_3 or data points residing near or inside dense cluster C_1?

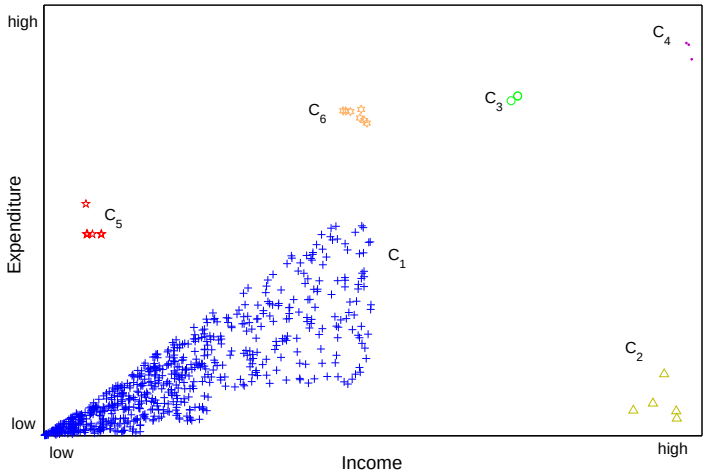


Figure 1.1: Objects in two dimensional space

In the above example if the objective is to find outliers using existing techniques such as distance based [47] or density based [17] then most likely these approaches will find data points belonging to the clusters C_2, C_3, C_4, C_5 and C_6 as highly ranked potential outliers. This is because these data points are isolated, far away from their k nearest neighbor, and hence are easily detected as outliers. Intuitively, high expenditure when income is high as indicated by the data points in clusters C_4 should not be flagged as outliers. Similarly, data points forming cluster C_2 are not anomalies. Real outliers which "make sense" are the data points belonging to the clusters C_5 and C_6. The challenge is to overcome the mismatch between outliers as entities "which are far away from their neighbors" and "real" outliers.

One way to ascertain interestingness and usefulness of the reported outlier is by making use of domain knowledge. In this thesis, we present measures to discover outliers based on background knowledge, represented by a *Bayesian network*. A Bayesian network is a probabilistic graphical model for describing domain knowledge and reasoning under uncertainty. Bayesian networks use the graph metaphor to model (causal) interactions among set of variables, where the variables are represented as nodes of a graph and the interactions as directed links (also known as arcs or edges) between the nodes. The key characteristic of Bayesian networks is their ability to encode directional relations which can represent *cause-effect* relationships, compared to other

graphical models that cannot for example, Markov networks. It is common wisdom to consider a good Bayesian network as a causal model where, causality flows in direction of edges [48]. Their ability to represent domain knowledge in the form of directional relationships is an important reason for choosing them as a baseline to explore observations which violate causal relationships. Based on domain knowledge captured by a Bayesian network, we propose:

Definition: *Interesting anomalies are those data points which violate the causal semantic captured via a Bayesian network.*

1.2 Why Bayesian Networks?

In this thesis we focus on use of Bayesian networks to capture domain knowledge in order to mine interesting anomalies for the problem domain. There are several reasons which have led us to this approach. First and foremost, we needed a knowledge representation model which can encode domain knowledge based on which we could discern between "interesting" and "uninteresting" anomalies. Bayesian networks provide a *declarative representation* of knowledge for the domain. The key property of a declarative representation is the separation of knowledge from the reasoning process [48]. This implies, once Bayesian network is designed with domain knowledge, reasoning on whether a new data point is an anomaly can be employed without updating the model. Second, compared to other graphical models such as Decision trees and Markov models, only Bayesian networks can represent causal relationships. These characteristics of Bayesian model can prove to be very useful in mining those observations which violate common causal knowledge encoded in the model. Third, Bayesian networks have a capacity to model the joint probability distributions (or JPD) compactly, which means that we can study each causal interaction encoded in the model independently. The JPD over all variables X_1, X_2...,$X_{|X|}$ is represented using chain rule as shown in Equation 1.1. Where notation $P(X_i \mid Pa(X_i))$ denotes probability of X_i given a set of its parent nodes denoted by $Pa(X_i)$. This special feature of BN is very useful to explain the reasons of unusual behavior of anomalies. Finally, Bayesian networks allow complex probabilistic queries to be performed on the model which is an advantage for mining

events which have either a high or a low probability to appear.

$$P(X_1, X_2...., X_{|X|}) = \prod_{i=1}^{|X|} P(X_i | Pa(X_i)) \tag{1.1}$$

Example: We show the advantages of using BN for anomaly detection in the following example. Figure 1.2 presents a small hypothetical Bayesian network. Suppose that we assume the knowledge encoded in the Bayesian network is complete and no factors other than smoking (variable Smoking) can influence the probability of cancer (variable Cancer) in a person. Furthermore, suppose both variables are binary in nature with states present (or P) and absent (or A). From the anomaly detection point-of-view, we may ask which observations are suspicious for this domain? We believe that the observations could be: (1) presence of cancer in absence of smoking and, (2) absence of cancer in presence of smoking. Both of these events could be anomalous because they contradict the knowledge encoded in the model, i.e., high probability in presence of cancer can be incurred only when it is known that a person smokes and has a lower chance of cancer in absence of smoking.

Figure 1.2: Bayesian network representing causal interaction between Smoking and Cancer

Figure 1.3: Bayesian network representing causal interaction between Smoking, Cancer and X-ray

The causal dependency between Smoking and Cancer can help discover events which under the domain knowledge are less likely to appear. The probabilistic queries such P(Cancer = P | Smoking = A) or P(Cancer = A | Smoking = P) could be answered to reveal if the event has high or low probability. Now suppose we extend the BN modeled in Figure 1.2 by joining a variable X-ray with variable Cancer (refer Figure 1.3). In this case, two causal relationships, (Smoking, Cancer) and (Cancer , X-ray) can be independently studied for anomalies due to Bayesian JPD and conditional independence

property. By understanding each causal interaction, we can potentially discover causal relationship in which anomalies are present.

1.3 Contributions

In this thesis we present outlier detection approaches based on domain knowledge captured by a Bayesian network models specifically for categorical and numerical data sets. We summarize our contributions as follows.

1. In order to discover "real" and "interesting" anomalies the integration of domain knowledge into the discovery process is required. The domain specific knowledge here can be treated as a model against which the data points violating common encoded knowledge are reported as "real" outliers. In this thesis, the use of Bayesian networks is proposed to capture domain knowledge. Bayesian networks provide visualization of causal interactions among attributes that exist in the domain. By exploiting these causal interactions, we can not only discover interesting anomalies but, are also able to provide contextual information for the discovered anomaly. We propose solution techniques for identifying anomalies and providing an explanation for the discovered anomalies in data sets which contain both categorical and numeric attributes.

2. We propose a novel algorithm which combines the use of Bayesian network and probabilistic association rules to discover and explain anomalies in categorical data. The Bayesian network allows us to organize information in order to capture both correlation and causality in the feature space, while the probabilistic association rules have a structure similar to association mining rules. In particular, we focus on two types of rules: (i) *low support & high confidence* and, (ii) *high support & low confidence*. New data points which satisfy either one of the two rules conditioned on the Bayesian network are the candidate anomalies. Extensive experiments performed on well-known benchmark data sets demonstrate that our approach is able to identify anomalies with high precision and recall over existing traditional outlier detection techniques. Moreover, our approach can be used to discover contextual information from the mined anomalies, which other techniques often fail to do so.

3. Outliers are often identified as data points which are "rare", "isolated", or far away from their nearest neighbours. We demonstrate that meaningful outliers, i.e., outliers which perhaps encode important or new information are those which violate causal relationships. A critical analysis on distance based techniques is presented which highlights why distance based criteria may not be an accurate and effective technique to discover true outliers using real life examples and data sets. We provide clear evidence that our approach, which uses Bayesian Networks, has the potential to discover real outliers.

4. We present a new measure to discover outliers in numerical data sets and also data sets containing a mixture of data types by using a Gaussian Bayesian network and a Hybrid Bayesian network respectively. Data points which violate the causal relationships encoded in these two forms of networks are reported as outliers. Several experiments performed confirm that outliers identified in this fashion are in some sense "genuine" as they reveal new information about the underlying data generating process.

1.4 Organization

This thesis is structured as follows. In Chapter 2, we introduce related work in research of outlier detection techniques and highlight their key advantages and disadvantages. The theoretical concepts of Bayesian networks, inference in Bayesian networks, Bayesian structure and parameter learning and some leading Bayesian softwares and packages are discussed in Chapter 3. In Chapter 4, we present novel anomaly detection technique which combines use of Bayesian network and probabilistic association rules to discover and explain anomalies in categorical data. A critical analysis on Bayesian and distance based approaches for anomaly detection is also presented. A measure for anomaly detection using Gaussian and Hybrid Bayesian networks is described in Chapter 5 where, based on causal relationships encoded in the model, we identify those points as outliers which violate these causal relationships. We conclude in Chapter 6 with a summary of the thesis, and directions of future research work.

In addition to six chapters detailed above, this thesis contains one Appendix. For all Bayesian networks shown in Chapters 4 and Chapter 5, list of attributes encoded in the models are listed in Appendix A.

Chapter 2

Background

Anomaly or outlier detection aims for discovering patterns in data that do not conform to a normal or expected behavior. The term normal refers to a baseline that may be known a priori or learned through time. The presence of outliers in a data set may be due to noise or unwanted system behavior. Noise may be caused by measurement error or communication error, but the nature of unwanted system behavior is application dependent. For example, in network or system performance monitoring, it may be link or server failures, and in security, it may be denial of service attacks or intrusion detection. In accounting and transaction monitoring, it may be due to fraud, whereas in surveillance applications, it may be due to abnormal activity. The approaches used to perform anomaly detection depend on the application and the nature of the data.

In this chapter we discuss existing anomaly detection techniques under three main categories namely *general*, *contextual* and *Bayesian network* based approaches as shown in Figure 2.1. General techniques which include distribution, distance and density are ones which are oldest in the literature of outlier detection. The key characteristics of distance and density based techniques are that they do not require prior knowledge of the application domain in order to mine outliers, and are well suited for numerical and categorical data sets. However, distribution-based approaches may require prior knowledge about distribution of data to discover outliers. In contrast to general techniques, contextual anomaly detection techniques aim for data points which are anomalous in some context but not otherwise. These techniques do require domain knowledge before discovering process. Similar to contextual techniques for outlier detection, Bayesian network based techniques require to model domain knowledge. Under Bayesian setting, anomalies are often those observations which are unseen or low probable.

In addition to this categorization, there exist several variants which focuses in discovering anomalies from spatial and sequential data sets. Sun, Chawla and Arunasalam proposed spatial anomaly detection techniques in climate data and sequential anomalies in protein sequences [68; 20; 69].

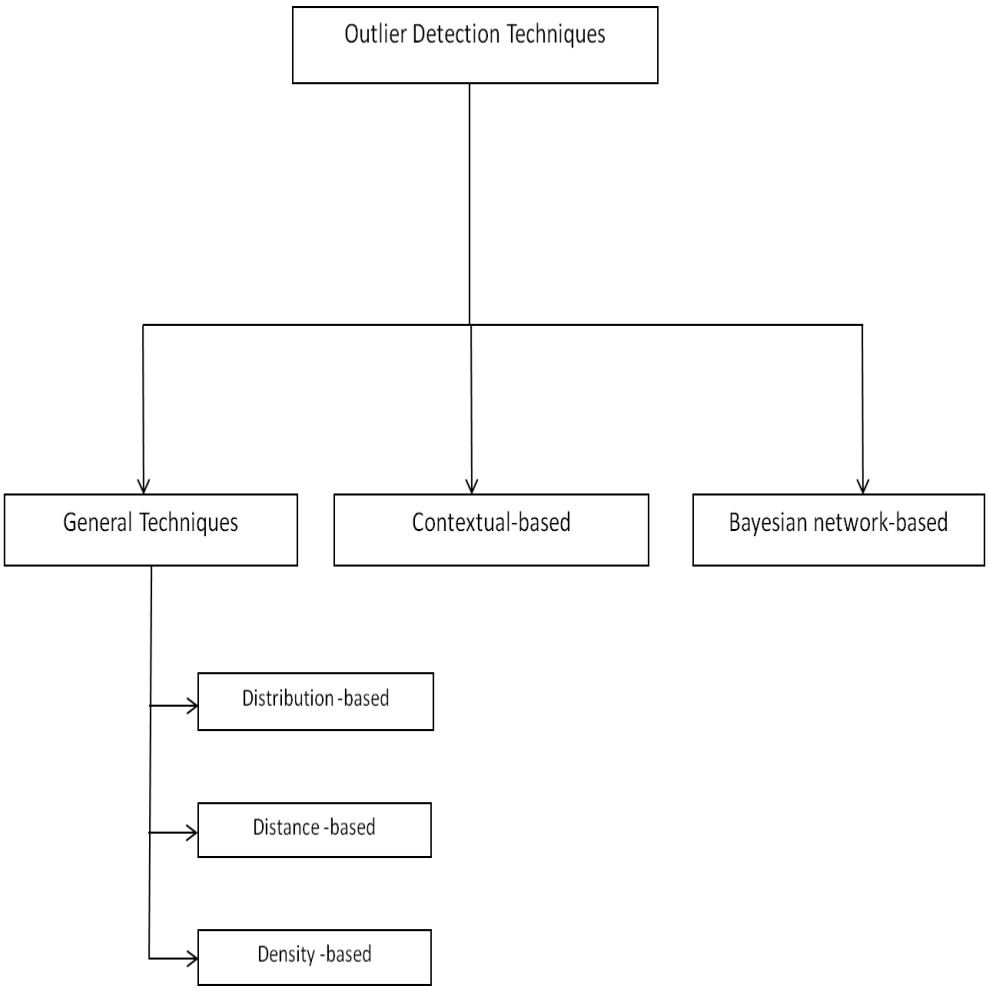


Figure 2.1: Broad categories of outlier detection techniques

2.1 Distribution-based Approaches

The problem of outlier detection has been extensively studied in the statistical community and is perhaps the oldest approaches. Definitions proposed by Hawkins [37],

Barnett and Lewis [11] formed bases of outlier detection in statistics. Below we present their classical definitions of outliers.

Definition 1: *an outlier is an observation that "deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism"* [37].

Definition 2: *an outlier is an observation that is "numerically distant from the rest of the data"* [11].

Consider the following sample data:

-0.91, -0.62, -0.87, -0.78, -0.61, -0.68, -0.77, ***-67.91****, -0.99, -0.81, -0.73, -0.71*

On inspection, without priori knowledge we can identify observation -67.91 as an outlier since its value is very distinct from rest of the data. Distribution based approaches can further be categorized as those meant for univariate data, i.e., where observations are of single variable and those designed for multivariate data containing multiple variables. We discuss each of the case below.

2.1.1 Univariate outliers

The most basic type of outlier detection is called univariate outlier detection, where observations are of a single variable. In this setting, we are given a set of observations (that are single values) and aim is to identify observations that are very far away from the other observations. Suppose we are given set of n observations $X = (x_1, x_2, ..., x_n)$. In order to measure how far away any observation is from the rest of the data, measures *mean* and *standard deviation* are used. Using these measures, outlying observations are identified using Grubbs' test or maximum normalised residual given by Equation 2.1. Where μ represents mean over n set of observations X and σ is the standard deviation. This simple test can be used to detect more than one outlier in a data set by applying it iteratively, removing one observation every time. However, this simple test is unreliable in the case if data contains large outliers. Reason being, large data values will distort

the sample mean and standard deviation and hence the test.

$$G = \frac{|x_i - \mu|}{\sigma} \tag{2.1}$$

The Grubbs test described above illustrates the challenge of detecting univariate outliers in observations that come from an unknown distribution. However, there are tests for example *student's t-test* which is used in the cases where it is assumed that data follows a normal distribution. As an another simple test which also assumes a normal distribution is by declaring all data instances that are more than 3σ distance away from the distribution mean μ, where σ is the standard deviation for the distribution as outliers. Such definition of outliers has application in for example, quality control domain [65]. In cases where distribution of data are not known at priori, Chebyshev's inequality theorem defined in Equation 2.2 is used by outlier detection techniques. where x_i represents the data, μ is the data mean, σ is the standard deviation of the data, and k represents the number of standard deviations from the mean. Chebyshev's inequality gives a lower bound for the percentage of data that is within a certain number of standard deviations from the mean, not dependent upon how the data is distributed.

$$P(|x_i - \mu| \leq k\sigma) \geq \left(1 - \frac{1}{k^2}\right) \tag{2.2}$$

An another method to analyse data for outliers for unknown distribution are due to [73], who invented the boxplot as a way to visualize and explore data. An example of boxplot is shown in Figure 2.2. The boxplot graphically depicts five number summary: minimum (min), first quartile (Q_1), median, third quartile (Q_3) and the maximum (max). In the simplest box plot the central rectangle spans the first quartile to the third quartile called as interquartile range or IQR. In this setting, outliers are observations either ($3 \times$ IQR) or more above the third quartile or ($3 \times$ IQR) or more below the first quartile.

2.1.2 Multivariate outliers

In multivariate data, outlier detection becomes a slightly less intuitive problem because it is not as obvious what is considered far away or atypical when observations are composed of more than one variable. Observations of more than one variable introduce new complexity into the outlier identification problem because in multivariate data it is

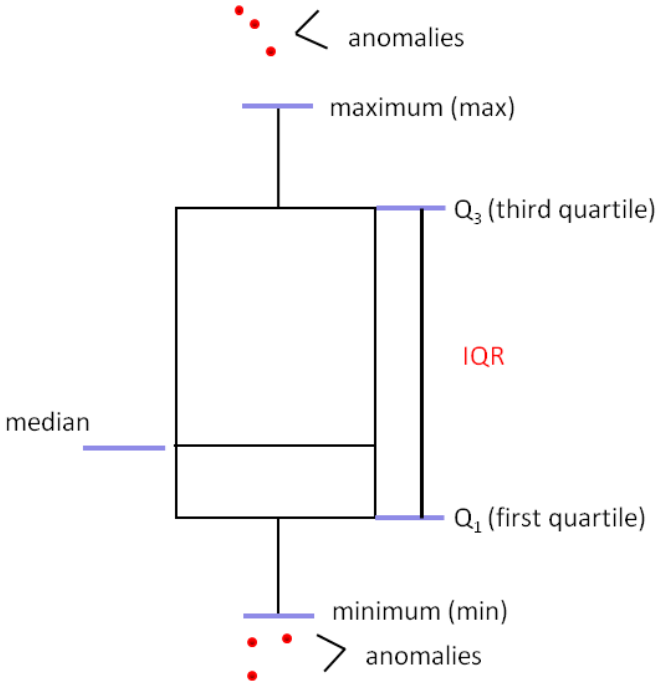


Figure 2.2: Boxplot

necessary to take into account not only the individual variables, but also the interactions of these variables. Take, for instance, the data shown in Figure 2.3. This figure shows observations consisting of two variables (Var 1 and Var 2). There are two clear outliers visible in this figure (the red data point), yet these points are not outliers in either direction (Var 1 or Var 2).

Mahalanobis[52] distance measure takes into account covariance that exist between pair of variables in order to discover anomalies from a multivariate data. Let A be a (m $\times$ n) matrix of observations where the row in A are the observations and the columns of A are the variables.

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

Mahalanobis distance is calculated for each observation as defined in Equation 2.3. Where $\bar{x}$ is the center of the data estimated as a vector whose columns are the means of the individual variables. The Σ^{-1} denotes the inverse of the $(n \times n)$ covariance

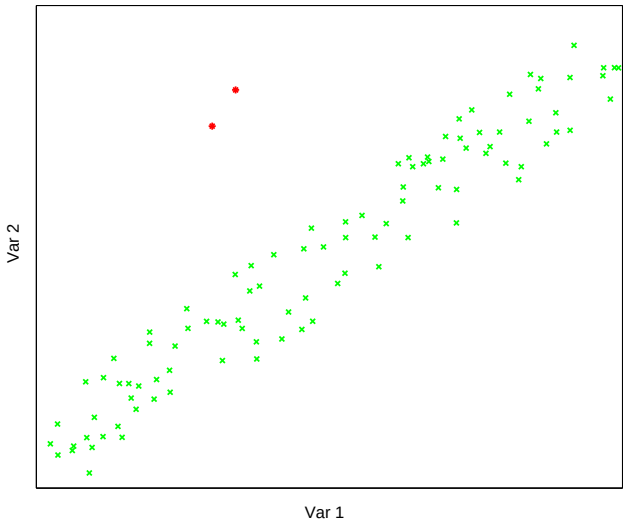


Figure 2.3: Shows two data points visibly outliers but not outliers in either direction of Var 1 or Var 2

matrix. The covariance between pair of variables is calculated using Equation 2.4. An observation with a large Mahalanobis distance is considered as an outlier. Assuming that the data follows a multivariate normal distribution, square of Mahalanobis follows a Chi-Square distribution for a large number of instances. Therefore the proposed cutoff point in Equation 2.3 is given by χ^2_n, where χ^2 stands for the Chi-Square distribution for n dimensional data with signification level usually taken as 0.05.

$$D_i = \sqrt{(x_i - \bar{x})^T \Sigma^{-1} (x_i - \bar{x})} \tag{2.3}$$

$$cov(X,Y) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \tag{2.4}$$

2.1.3 Advantages and Disadvantages of Distribution-based Techniques

The advantages [74] [54] of distribution based techniques are as follows:

1. If the distribution estimation step is robust to anomalies in data, statistical techniques provide a statistically justifiable solution for anomaly detection.

2. If the underlying assumption of distribution of data hold true, statistical based techniques can operate in an unsupervised setting without need of labels to evaluate results.

The disadvantages [74] [54] of distribution based techniques are as follows:

1. Statistical methods for anomaly detection usually depends on arithmetic mean and technique may completely fail in case mean is outlier, refer Figure 2.4

2. This method suffers from what is described as “curse of dimensionality” [13].

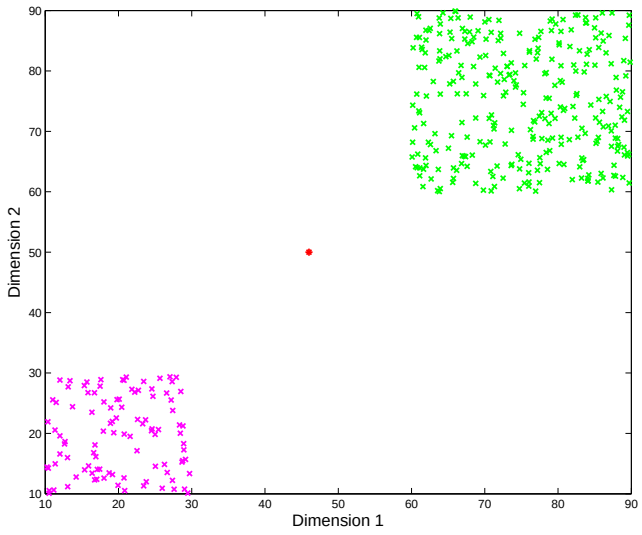


Figure 2.4: Shows mean as outlier

2.2 Distance-based Approaches

In distance based approaches for anomaly detection, each data point is analysed with respect to its nearest neighbor. The basic assumption followed by these techniques is that the *normal instances occur in dense neighborhood while, anomalies occur far from their closest neighbors* [74]. We introduce two commonly available definitions of distance based approaches below proposed by Knorr and Ng [45] and Ramaswamy et al. [60].

Definition 1: *Let object O in a data set T is a DB(p, D)-outlier if at least fraction p of the objects in T lies greater than distance D from O* [45].

Definition 2: *Given an input data set with N points, parameters n (total number of outliers we are interested in) and k (number of neighbors of a point that we are interested in), a point p is a D_n^k outlier if there are no more than n-1 other points p' such that $D^k(p') > D^k(p)$* [60].

The main idea proposed by Knorr and Ng in their research of outlier detection [45; 46; 47] was to compute the anomaly score of a data instance by counting the number of nearest neighbors that are not more than d distance apart from the given instance. The distance-based approaches like proposed by Knorr and Ng require a distance or a similarity measure defined by two instances. Distance between two data instances is generally computed using Euclidean and Hamming measures depending on whether variables are numerical or categorical. Ramaswamy et al. [60] simplified the definition of outlier by stating that outliers are the top n data points whose distance to the k^{th} nearest neighbor is greatest. Consider Figure 2.5 where some random data points are shown and let k (number of nearest neighbor) is set to 2. Data points O_1 and O_2 are two outliers with distance to their 2^{nd} nearest neighbour are largest. We now present two classical distance based algorithms namely, Nested-Loop and Nested-Loop with randomization and pruning algorithm.

2.2.1 Nested-Loop Algorithm

Nested-Loop algorithm is a very simple approach of mining outliers where a sequential computation of distance between every two data points is performed until k neighbors within distance D are found. If for each object, k neighbors within distance D are found then the data point is not an outlier else it is marked as outlier. The worse case complexity of this method is $O(dN^2)$ where, d and N are the dimensionality and size of data set.

The Nested-Loop algorithm uses a block oriented design. Supposing buffer size of B% of the data set size is given. The algorithm then divides the entire buffer space in two halves called first and second arrays. It reads the data set into the arrays, and

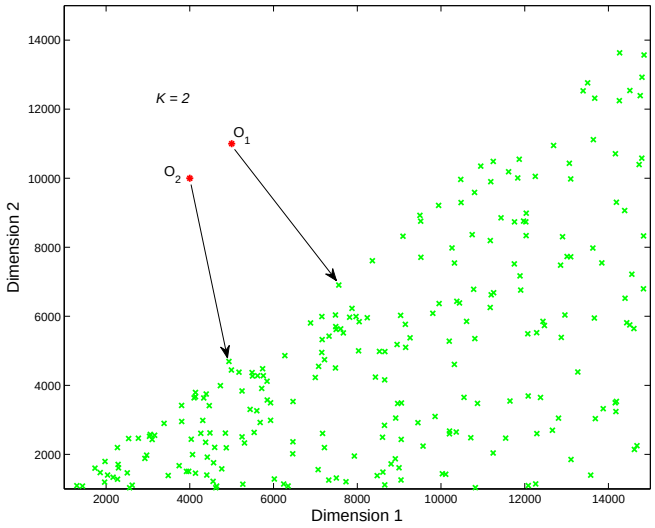


Figure 2.5: Shows top two outliers O_1 and O_2 with distance to their 2^{nd} neighbor being largest

directly computes the distance between each pair of objects. For each object O in the first array, a count of its k neighbors is maintained. Counting stops for a particular object whenever the number of k neighbors exceeds D. The pseudo-code of the algorithm can be found in [69] [45].

2.2.2 Nested-Loop with Randomization and a Punning Rule

Bay and Schwabacher [12] proposed an algorithm for finding outliers by calculating distance from its nearest neighbors using nested loops in conjunction with randomization and pruning rule. The methodology presented gives near linear time performance on many large data sets having continuous and discrete features. The main idea in the algorithm presented is for each data point o; a track of its k^{th} closest neighbors is made, as the data set is scanned. When a data point (the k^{th} closest neighbor) has a distance less than cutoff threshold, the data point is no longer outlier and the next data point is tracked. As more and more data points are processed the cutoff increases along with the pruning efficiency.

The major strength of this algorithm is its near linear time performance as compared to quadratic performance of algorithm based on a nested loop nearest neighbor search. However, algorithm suffers from few limitation such as its dependence on parameters

block size and k which has a considerable effect on discovering outliers. Also, the algorithm assumes that the data is in random order and if data set is in sorted then performance could be poor. The pseudo-code of the algorithm can be found in [59] [12].

2.2.3 Advantages and Disadvantages of Distance-based Techniques

The advantages of distance-based [74] techniques are as follows:

1. Distance-based techniques are purely data driven. These techniques are unsupervised in nature and do not make any assumption about the underlying distribution of the data.

2. These techniques can be applied directly on data sets containing variables of different data type by just changing the distance metric.

The disadvantages of distance-based [74] techniques are as follows:

1. Ascertaining quality of reported outliers is really challenging for distance based techniques. There could exist cases where normal instances in data set may not have enough k nearest neighbors or cases where anomalies satisfy nearest neighbor condition.

2. Defining distance metric for complex data sets for example, graphs and sequences could be challenging.

2.3 Density-based Techniques

Density-based methods estimate the density distribution of the input space and then identify outliers as those lying in regions of low density. Such techniques estimate the density of the neighbourhood of each data instance. An instance that lies in a neighborhood with low density is declared to be an outlier while an instance that lies in a dense neighbourhood is declared to be normal. The distance-based outlier techniques discussed in the last section captures *global outliers*, because these definitions take a global view of the data set. For a data set with a simple structure, for example, one that contains one or more clusters with a similar density, these definitions work well. However, for many data sets that have a complex structure with regions of differing

density, the methods based on these two definitions may not be able to find all interesting outliers. Density based approaches are well suited in this scenario. Such techniques captures how isolated an object is with respect to its surrounding neighborhood rather than the whole data set. Outliers targeted by this approach is called as *local outliers*.

To illustrate this, consider visualization of data set in Figure 2.6. This data set contains two main clusters, one dense and one sparse. For two separated objects O_1 and O_2, which stay far away from the clusters are clear anomalies from both global and local view. However, for a distance based approach, data points O_3 and O_4 lying close to dense cluster will have distances approximately equal to any distance between two points in the sparse cluster and hence may fail to detect them. Therefore, an outlier detection method that takes into account local density variations is necessary to solve this problem.

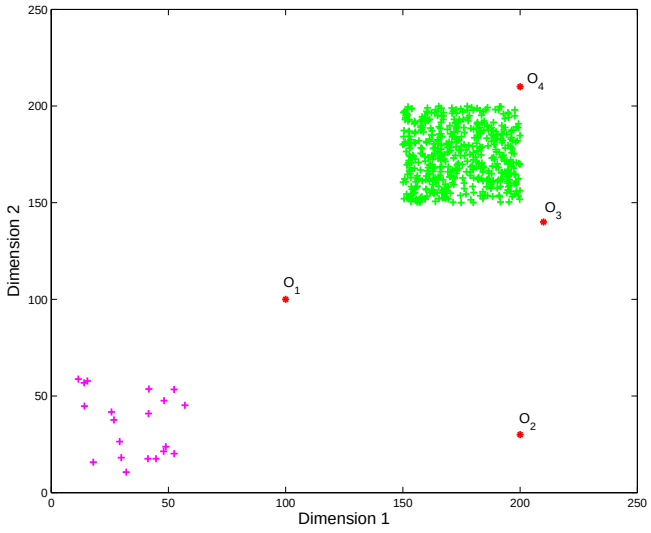


Figure 2.6: Shows one dense cluster, one sparse cluster and four outliers

2.3.1 Local Outlier Factor

A popular density-based approach, Local Outlier Factor (LOF) was originally proposed by Breunig et al. [17]. The LOF is computed for each object in the data set, indicating its degree of outlierness. This quantifies how outlying an object is. The outlier factor is local in the sense that only a restricted neighborhood of each object is taken into account. The LOF of an object is based on the single parameter called MinPts, which is

the number of nearest neighbors used in defining the local neighborhood of the object. The LOF of an object p can be defined by Equation 2.5

$$LOF_{MinPts}(p) = \frac{\sum_{o \in MinPts(p)} \frac{lrd_{MinPts(o)}}{lrd_{MinPts(p)}}}{|N_{MinPts}(p)|} \tag{2.5}$$

The outlier factor of object p captures the degree to which we can call p an outlier. It is the average of the ratio of the local reachability density of p and those of p's MinPts-nearest neighbors. The lower p's local reachability density (lrd) is, and the higher lrd of p's MinPts-nearest neighbors are, the higher is the LOF value of p. The local reachability density (lrd) of an object p is the inverse of the average reachability distance (reach-dist) based on the MinPts nearest neighbors of p. The local density can be ∞ if all the reachability distances in the summation are 0. This may occur for an object p if there are at least MinPts objects, different from p, but sharing the same spatial coordinates, i.e., if there are at least MinPts duplicates of p in the dat set. Local reachability density is defined by Equation 2.6. The reachability distance of an object p with respect to object o is defined by Equation 2.7.

$$lrd_{MinPts}(p) = \left(\frac{\sum_{o \in MinPts(p)} reach - distMinPts(p,o)}{|N_{MinPts}(p)|} \right) \tag{2.6}$$

$$reach - dist_{MinPts}(p,o) = \max MinPts(o), dist(p,o) \tag{2.7}$$

For any positive integer k, the k-distance of object p, denoted as k-distance(p), is defined as the distance d(p, o) between p and an object $o \in$ D where D is a data set such that:

1. for at least k objects $o' \in$ D | p it holds that d(p, o') $\leq$ d(p, o), and
2. for at most *(k-1)* objects $o' \in$ D | p it holds that d(p, o') $\leq$ d(p, o)

Figure 2.7 illustrates the idea of reachability distance with k = 4. Intuitively, if object p is far away from o (e.g., p_2 in the figure), the reachability distance between the two is simply their actual distance. However, if they are sufficiently close (e.g., p_1 in the figure), the actual is replaced by k-distance of o. The reason is that, the statistical fluctuation of d(p, o) for all the p's close to o can be significantly reduced. The strength

of this smoothing effect can be controlled by the parameter k. The higher the value of k, the more similar the reachability distances for objects within the same neighborhood.

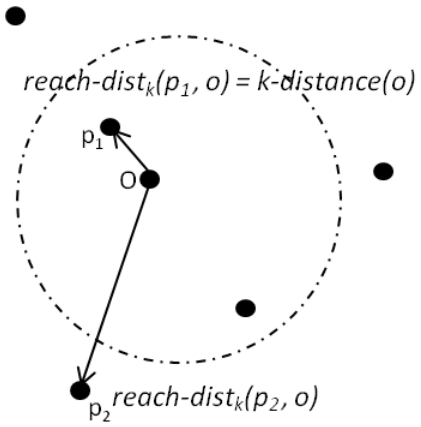


Figure 2.7: Reachability distance

2.3.2 Advantages and Disadvantages of Density-based Techniques

The advantages of density-based [74] techniques are as follows:

1. Density-based techniques are purely data driven. These techniques are unsupervised in nature and do not make any assumption about the underlying distribution of the data.

2. These techniques gives advantage on data sets carrying densities of data points in varying sizes for capturing local outliers.

The disadvantages of density-based [74] techniques are as follows:

1. Ascertaining quality of reported outliers is challenging as in the case in distance based techniques.

2. The major drawback is the computational complexity which is $O(N^2)$.

2.4 Contextual-based Techniques

Anomaly detection techniques such as distribution, distance and density based discussed so far in this chapter discovers *point based anomalies*. All of these techniques

consider an individual data instance as anomalous with respect to the rest of the data. However, there could exist data points which are anomalous in a specific context but not otherwise. The notion of context in data is induced by domain knowledge or by the structure of data. Figure 2.8 taken from Chandola et al. [74] explains one example of contextual outliers for a temperature time series. Figure shows the monthly temperature of an area over last few years. A temperature of 35F might be normal during winter (at time T_1) at that place, but the same value during summer (at time T_2) would be an anomaly.

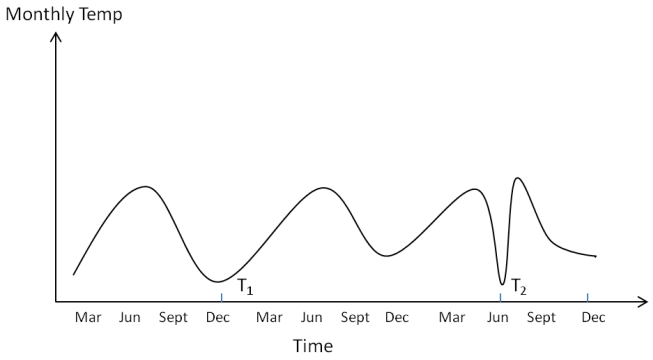


Figure 2.8: Shows T_2 as a contextual anomaly

Contextual outliers were addressed as *conditional outliers* by Song et al. [66]. They proposed a method called *conditional anomaly detection* (CAD). The emphasis of CAD methodology was to present user with anomalies which are interesting. For this they integrated domain knowledge in the discovering process. They suggested to divide attribute set of data into two categories namely environment and indicator attributes. Environment attributes were precisely those attributes which accounts for trends in the data, such as, season and temperature. The reminder of data attributes was called indicator attributes. We now discuss example from [66] for understanding on environment and indicator attributes. Consider Figure 2.9 where two variables namely, Max_daily_temp and Num_fever are monitored to detect disease outbreak at the earliest possible instant. The first variable tells the maximum outside temperature on a given day while, Num_fever tells how many people were admitted to a hospital emergency room complaining high fever. In this example, Max_daily_temp is the environment attribute and, Num_fever is the indicator attribute. In the Figure, data point O_1 and O_2 are both anomalies based on most conventional definitions. However, if it is considered that Max_daily_temp is not directly indicator of anomaly but, instead is responsible in

bringing trends then, data point O_2 is not an anomaly. The reason being, encountering large number of fever cases in a cold day is normal reducing interest in data point O_2. Whereas, data point O_1 is interesting from an anomaly prospective since it signifies the situation where there are number of people admitting in hospital with fever in a warm temperature.

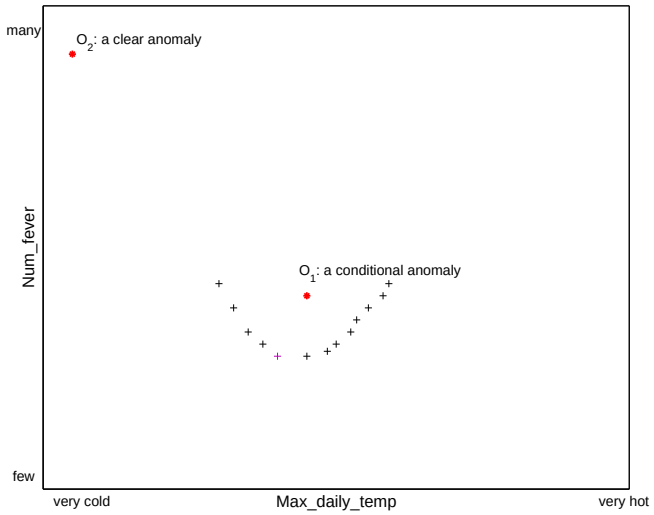


Figure 2.9: Shows O_2 as a clearest anomaly whereas, O_1 is a conditional anomaly

By forming the relational model between environment and indicator attributes, a data point was flagged anomaly depending on how much its indicator attribute values differ from the usual indicator attribute values. The three expectation maximization based [26] learning algorithms were proposed in order to learn the dependency model between two sets of attributes. In their work maximum likelihood estimation (MLE) was used to fit a multidimensional data set to model described using the probability density function for Gaussian Mixture Model. However, parametric distribution called f_{CAD} in their work does not treat all attributes identically. The f_{CAD} had the form f_{CAD} (y $\mid \Theta$, x) where, x is the set of environment attributes values, and y the set of indicator attribute values. This implies that a data point's environment attributes x are taken as input, and used along with the model parameter Θ to generate the set of indicator attributes y. By doing this, it is learnt how the environmental attributes map to the indicator attributes. A data point is considered anomalous against the model learnt if its indicator attributes cannot be explained in the context of environment attributes.

Contextual outliers have also been explored in time-series and spatial data [16] [15].

2.4.1 Advantages and Disadvantages of Contextual-based Techniques

The advantage of contextual-based techniques is as follows:

1. The key advantage of contextual anomaly detection techniques is that they allow a natural definition for an anomaly in many real life application where data instances tend to be similar within a context.

The disadvantage of contextual-based techniques are as follows:

1. The disadvantage of contextual anomaly detection techniques is that they are applicable only when a context can be defined. It may be difficult to define contextual knowledge for every application domain.

2.5 Bayesian Network-based Approaches

As an expert system Bayesian networks have been used in various fields. Pearl et al. [43] developed CONVINCE, an interactive decision-aiding expert system for situation assessment tasks. It was designed to help user in articulating ill-defined situation assessment problems in a formal structure, guides the user in searching for relevant information and data, and then deduces rational inferences from the formal structure. Munin [5], a medical expert system for diagnosing neuromuscular diseases is an another example of Bayesian application. Few more examples of usefulness and application of Bayesian networks are: Pathfinder [39], an expert system that assists surgical pathologists with the diagnosis of lymph-node diseases and win95pts [23], an expert system for printer troubleshooting developed by Microsoft. The reason of Bayesian popularity is because of its graphical representation and strong statistical inference mechanism. However, these studies consider only the essential ideas about the structure and use of Bayesian networks.

In other applications, Bayesian networks have been used for anomaly detection. Barbara et al. [10], Sebyala et al. [64], Mingming [78] and Bronstein et al. [18] proposed a network intrusion detection system based on Bayesian networks. For novelty detection in video surveillance an approach was proposed by Diehl and Hampshire [27].

Bayesian approach based rare event prediction in sensor data was proposed by Seong et al. [21]. Wong et al. [75] presented a technique for disease outbreak detection.

A general approach of Bayesian based anomaly detection is in multi-class setting. It uses a naive Bayesian network to estimate the posterior probability of observing a class label (from a set of normal class labels and the anomaly class label), given a test data instance. Bayesian network under this setting is trained on different classes, and then a trained model is used to test a new observation with largest posterior is chosen as the predicted class. Bayes estimator based network intrusion detection technique proposed in [10] is based on anomaly detection system called Audit Data Analysis and Mining(ADAM). ADAM applies association rule mining technique to look for the abnormal events in the network data, and then a classification algorithm was used to classify the abnormal events into normal or abnormal instances. Since knowledge on classifier is restricted to nature of data present in the training data so, to avoid this limitation, authors proposed pseudo-Bayes estimator as a means to estimate prior and posterior probability of new attacks. Janakiram et al. [25] presented a technique based on Bayesian network to identify local outliers in streaming sensor data. This technique used Bayesian network to capture not only the spatio-temporal correlations that exist among the observations of sensor nodes but also conditional dependence among the observations of sensor attributes. Each node trains a Bayesian network to detect outliers based on behaviors of its neighbor readings as well as its own reading. An observation is considered as outlier if it falls beyond the range of the expected class.

There exist several variants to the basic approach discussed above which used Bayesian networks for anomaly detection task. Examples of few studies are Seong et al. [21], Babbar et al. [9], Babbar and Chawla [7], Wong et al. [75], Babbar and Chawla [8],Cansado and Soto [19] and Wong et al. [75]. In all of these studies, Bayesian network has been modeled to capture background knowledge and anomalies were discovered in an unsupervised setting. Seong et al. [21] proposed a Bayesian network based on rare event prediction methods for high concentrations of high ozone O_3 forecasts in Seoul, Korea. Using expert knowledge, Bayesian network was modeled and boundary conditions using chemical reaction equations were established for parameters governing concentration of O_3. Babbar and Chawla [7] and Babbar et al. [8] proposed a technique for mining interesting anomalies based on background knowledge captured by a Bayesian network in categorical data sets. They exploited Bayesian causation and correlation encoded in the feature space using two probabilistic association rules in order to reveal anomalies

and to explain their anomalous nature. In [8], authors demonstrated that meaningful outliers, i.e., outliers which perhaps encode important or new information are those which violate causal relationships. The technique defined was especially designed for numerical data sets which used causal inference to reveal and explain anomalies.

Cansado and Soto [19] proposed an unsupervised approach for mining outliers in large databases using joint probability distribution (JPD) in Bayesian network. Using JPD records where ranked according to their oddness. Highly common records well explained by the model, received a high likelihood while, strange records received a low likelihood. To learn Bayesian structure, authors extended Sparse Candidate Algorithm [31] to able to use continuous variables.

Research contribution by Wong et al. [75] used a Bayesian network to model the domain for detecting disease outbreak. In their approach, attribute set was divided in user specified two groups, features those accounts for forming trends where grouped in environmental set whereas, attributes left formed indicator set. The Bayesian network used was conditioned on forming relation only between attributes belonging to environmental set to attributes in the indicator set. An algorithm WSARE 3.0 was developed to compare recent data against baseline distribution captured by the Bayesian network with the aim of finding rules that summarizes significant patterns of anomalies.

2.5.1 Advantages and Disadvantages of Bayesian network Techniques

The advantages of Bayesian network-based techniques are as follows:

1. Bayesian network-based approaches are designed on grounds of domain knowledge which gives an advantage of mining genuine outliers.

2. Using Bayesian theoretical concepts like conditional independence assumptions and joint probability distribution, we could explain why identified data point is an anomaly.

The disadvantage of Bayesian network-based techniques is as follows:

1. Generally a large amount of data is required in order to learn Bayesian networks in absence of domain experts.

2.6 Summary and Conclusion

In this chapter, we presented extensive overview of existing outlier detection methods. The methods discussed are closely related to or foundations of this thesis. Anomaly detection methods discussed in Sections 2.1, 2.2 and 2.3 discovers point based anomalies, i.e., these technique considers an individual data instance as anomalous with respect to the rest of the data. The advantages of these techniques are several. First and foremost, all these techniques are purely unsupervised except for few distribution based techniques discussed in Section 2.1 where an underlying distribution is assumed before process of anomaly detection is applied. Second, such techniques can be applied on data sets containing different data types by just changing the distance metric. Lastly, the notion of global and local outliers given by distance (Section 2.2) and density (Section 2.3) techniques are capable of capturing anomalies residing far from their neighbors, and those far from their local neighborhoods. The key limitation of these techniques is that they are largely focused on the design of measures and algorithms to identify outliers in large and high dimensional categorical and numeric databases. However, not much stress has been given on the interestingness of the reported outlier. In Chapter 5 and 6, we propose a Bayesian network approaches which claims to mine "genuine" and "meaningful" outliers.

We also discussed in Sections 2.4 and 2.5 anomaly detection approaches which are based on domain knowledge to discover anomalies. However, research in this area is limited. To best of our knowledge, studies which considers importance of not only identifying anomalies, but to also explain them is very limited. The techniques we address in Chapters 5 and 6 for anomaly detection are capable of both *discovering* and *describing* anomalies.

Chapter 3

Bayesian Network Models

The main focus of work in this thesis revolves around theory of *Bayesian networks*. Bayesian networks is a kind of graphical model used to capture domain knowledge for reasoning and decision making under *uncertainty*. In many problem domains uncertainty is caused due to availability of vague and incomplete knowledge about the system. As a result it may lead to inappropriate conclusions. Probabilistic theory, aided with the methods of statistical analysis in Bayesian network provides means of coping with the problem of uncertainty thus, help drawing conclusions which are possible. Probability forms foundations of Bayesian network theory, and hence we first discuss key concepts of probabilistic theory before proceeding to introduction on Bayesian networks.

The rest of the Chapter is organised as follows. Primarily concepts of probability theory are discussed in Section 3.1. In Section 3.2, we introduce Bayesian networks. Summary and basic notations followed in this chapter are presented in Section 3.3. Section 3.4 deals with foundations of Bayesian network. The general rule of flow of information in Bayesian models is discussed in Section 3.5. Section 3.6 is focussed on basic concepts involving continuous variables in Bayesian networks. Different kinds of probabilistic inference and variable elimination algorithm are discussed in Section 3.7. We briefly introduce Bayesian parameter and structure learning in Section 3.8. In Section 3.9, we discuss Bayesian networks as causal models. Finally, in Section 3.10, we briefly discuss popular Bayesian software and packages.

3.1 Probabilistic Theory

3.1.1 Probability and Events

Probability refers to the likelihood or relative frequency for something to happen. For example, the weather report might say "there is high probability of rain today". Probabilistic theory deals with such estimates and the rules which they should obey. Probabilities are assigned to *events* say α, which can be considered as an outcome of an experiment for example, a coin flip. The set of possible events for an experiment is addressed as *event space* and, is often denoted by symbol Ω. For example, in case of dice, we might set $\Omega = \{1, 2, 3, 4, 5, 6\}$. Given event space, probability distribution over events denoted by $P(\alpha)$ satisfies following three conditions:

1. For any event α, $0 \leq P(\alpha) \leq 1$
2. $P(\Omega) = 1$
3. For any two mutually exclusive events α and β the probability that either α or β occur is given by Equation 3.1

$$P(\alpha \; or \; \beta) = P(\alpha \vee \beta) = P(\alpha) + P(\beta) \tag{3.1}$$

Condition 1 simply says that probability is a non-negative real number less than or equal to 1. By condition 2 it is meant that all possible outcomes have the maximal possible probability of 1. Condition 3 states that if two events cannot co-occur then, the probability that either one of them occurs equals the sum of the probabilities of their individual occurrences.

3.1.2 Random variables

In a daily life, it is often more natural to consider attributes of the outcome. For example, a person might have attributes such as "age", "gender", "height" and many more. Formally, a *random variable* is relation of attributes to their values in different outcomes. Random variable can take different set of values depending on its type.

Broadly, random variable can take finite values (categorical, for example gender) or infinitely many values, integer or real (for example, height). We follow the convention of using upper case letters to denote random variables such as X, Y, Z. And, lower letters to denote generic value of a discrete random variable, i.e., P(X = x) denotes probability of random variable X in x. The notation Val(X) represent set of values discrete random variable X can take. For example, Val(gender) = {male, female}. If notation, |Val(X)| specifies total number of value X has then, Equation 3.2 holds. The distribution over such a variable is called a multinomial.

$$\sum_{i=1}^{|Val(X)|} P(X = x_i) = 1 \tag{3.2}$$

Unlike discrete random variables, continuous random variables can take infinite set of possible real numbers in $\Re$. Probability over continuous random variable X is defined as a probability density function (PDF), p : $\Re \rightarrow \Re$ if it is a nonnegative integral function which satisfies Equation 3.3. That is, the integral over the set of possible values of X is 1. The PDF defines a distribution for X as in Equation 3.4 for any x in event space. Continuous random variables can have simplest PDF from Uniform to more complex Gaussian distributions. A random variable X has a Gaussian distribution with mean μ and variance σ^2, denoted $N(\mu; \sigma^2)$, if it has the PDF defined in Equation 3.5

$$\int_{Val(X)} p(x)dx = 1 \tag{3.3}$$

$$P(X \leq a) = \int_{-\infty}^{a} p(x)dx \tag{3.4}$$

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.5}$$

A Gaussian distribution has a bell-like curve where the mean parameter control the location of peak and variance determines how the Gaussian peaked is. In Figure 3.1 PDF of two Gaussian distribution are shown.

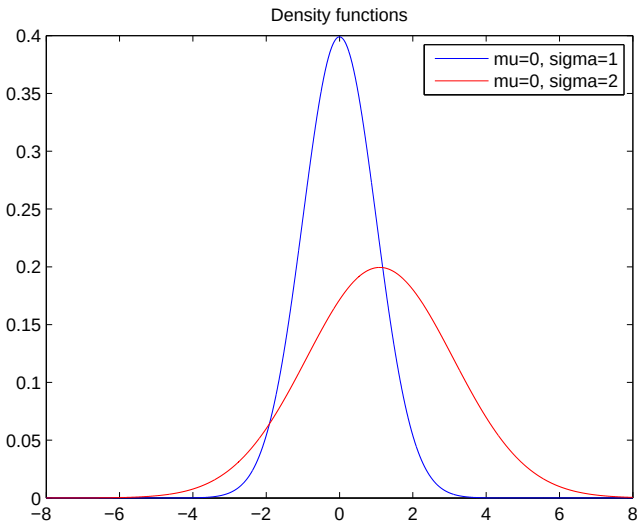


Figure 3.1: Example PDF of two Gaussian distributions

3.1.3 Conditional probability and Chain rule

Suppose we consider a distribution over a population of patients visiting a hospital. Here space of outcome is simply set of all patients in the population. Now, suppose we want to reason around the patient's test result (say event α) and possibility of a disease (say event β). That is, given test result, we are interested in knowing chances of a disease. More precisely, after learning that an event α is true, how do we change our probability about β occurring? Concept of conditional probability helps solving such queries. Formally, *conditional probability* on β given α is defined as in Equation 3.6.

$$P(\beta|\alpha) = \frac{P(\alpha \cap \beta)}{P(\alpha)} \tag{3.6}$$

Equation 3.6 can be rearranged as Equation 3.7. This equality is known as chain rule in probability. If suppose α_1, α_2,....,α_n are n events, then probability of these events can be expressed using Equation 3.8. Chain rule allows expressing probability of a combination of several events in terms of the probability of the first, the probability of second given the first and so on. In chain rule, order of events does not change the result, i.e., we can put events in any order. For three events, α_1, α_2, α_3, example of chain rule is presented by Equation 3.9

$$P(\alpha \cap \beta) = P(\alpha|\beta)(P(\beta) \tag{3.7}$$

$$P(\alpha_1 \cap .... \cap \alpha_n) = P(\alpha_1)P(\alpha_2|\alpha_1)....P(\alpha_n|\alpha_1 \cap ...\alpha_{n-1}) \tag{3.8}$$

$$P(\alpha_1 \cap \alpha_2 \cap \alpha_3) = P(\alpha_1)P(\alpha_2|\alpha_1)P(\alpha_3|\alpha_1, \alpha_2) \tag{3.9}$$

As with discrete random variables, conditional probabilities can be defined over continuous variables. Suppose, we are interested in defining P(Y | X = x). We could not use Equation 3.6 for P(X = x) = 0 which is not defined. To avoid this, conditioning on event (x - ε) $\leq$ X $\leq$ (x + ε) could yield positive probability [48]. We define in Equation 3.10 conditional probability when $\varepsilon \to 0$.

$$P(Y|X) = \lim_{\varepsilon \to 0} P(Y|X - \varepsilon \leq X \leq X + \varepsilon) \tag{3.10}$$

The limit in Equation 3.10 can be derived from a continuous joint probability p(x, y). Consider an event Y such that, a $\leq$ Y $\leq$ b. Equation 3.10 can be solved as below:

$$\lim_{\varepsilon \to 0} P(Y|X - \varepsilon \leq X \leq X + \varepsilon) = \frac{P(a \leq Y \leq b, x - \varepsilon \leq X \leq x + \varepsilon)}{P(x - \varepsilon \leq X \leq x + \varepsilon}) = \frac{\int_a^b \int_{x-e}^{x+e} p(x', y) dy dx'}{\int_{x-e}^{x+e} p(x') dx'} \tag{3.11}$$

In case ε is sufficiently small, approximation can be applied on Equation 3.11 by factor $\int_{x-\varepsilon}^{x+\varepsilon}$ p(x′)dx′ $\approx 2\varepsilon$ p(x). Similarly factor:

$$P(a \leq Y \leq b|x - \varepsilon \leq X \leq x + \varepsilon) \approx \frac{\int_a^b 2\varepsilon(x, y) dy dx'}{2\varepsilon p(x)} = \int_a^b \frac{P(x, y)}{p(x)} dy \tag{3.12}$$

Thus, $\frac{p(x,y)}{p(x)}$ is the density of P(Y | X = x).

3.1.4 Bayes' rule

Using symmetry property, Equation 3.7 can be rewritten as 3.13. Further, Equation 3.13 derives Bayes' rule represented by Equation 3.14.

$$P(\alpha \cap \beta) = P(\beta|\alpha)(P(\alpha) \tag{3.13}$$

$$P(\alpha|\beta) = \frac{P(\beta|\alpha)(P(\alpha)}{P(\beta)} \tag{3.14}$$

It is common to think of Bayes' rule in terms of updating our belief about a hypothesis α in the light of new evidence β. Specifically, posterior belief P($\alpha \mid \beta$) is calculated by multiplying our prior belief P(α) by the likelihood P($\beta \mid \alpha$) that β will occur if α is true. The power of Bayes' rule is that in many situations where we want to compute P($\alpha \mid \beta$) it turns out that it is difficult to do so directly, yet we might have direct information about P($\beta \mid \alpha$). Bayes' rule enables us to compute P($\alpha \mid \beta$) in terms of P($\beta \mid \alpha$).

For example, suppose that we are interested in diagnosing cancer in patients who visit a clinic. Let α represent the event *person has cancer*, and let β represent the event *person is a smoker*. On the basis of past data, let we have prior probability of events, P(α) = 0.1 and, P(β) = 0.5. Suppose, We are interested in knowing probability of person suffering from cancer given he smokes, i.e., P($\alpha \mid \beta$)? It is difficult to solve this query directly. However, we are likely to know P($\beta \mid \alpha$) from records specifying the proportion of smokers among those diagnosed. Suppose P($\beta \mid \alpha$) = 0.8. Based on this information, we can compute our query using Bayes' rule, refer Equation 3.15. Thus, in the light of evidence that the person is a smoker we revise our prior probability from 0.1 to a posterior probability of 0.16.

$$P(\alpha|\beta) = \frac{(0.8 \times 0.1)}{0.5} = 0.16 \tag{3.15}$$

Application of Bayes' rule is central to inference in Bayesian network which we will discuss shortly.

3.1.5 Joint probability distribution and Marginalization

Unlike conditional probabilities which are used to determine how much the occurrence of one event influences the occurrence of another event, *joint probabilities* determines the likelihood of two separate events simultaneously. The joint probability for two events, α and β, is expressed mathematically as P(α ,β). Joint probability is calculated by multiplying the probability of event α, expressed as P(α), by the probability of event β, expressed as P(β). Consider example of diagnosing cancer in patients taken above. Let α and β represent events *person has cancer* and *person is a smoker* respectively.

The joint probability of these events would be 0.05.

In comparison to joint probabilities, *marginalization* helps computing probability of subset of events from Ω. For example, we might want to compute P(X) from a joint distribution P(X, Y, Z). This is computed by summing over all possible combinations of values Y and Z to solve P(X). For example, suppose Val(Y) = (y_1, y_2) and, Val(Z) = (z_1, z_2). Then, P(X) can be computed from the joint distribution as below:

$$\begin{aligned} P(X,Y,Z) = \quad & \sum_{Y}\sum_{Z} P(X,Y,Z) = \\ & P(X,Y=y_1,Z=z_1) + P(X,Y=y_1,Z=z_2) \\ & + P(X,Y=y_2,Z=z_1) + P(X,Y=y_2,Z=z_2) \end{aligned} \tag{3.16}$$

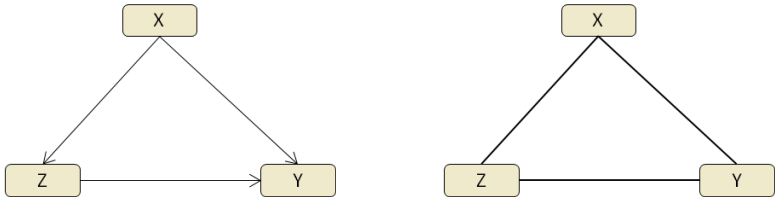
3.2 Introduction to Bayesian Networks

Bayesian network (BN for short) is a kind of probabilistic graphical model which combines *probability* with *graph theory* to compactly represent real world problems. Probability gives advantage of dealing with "uncertainty". In many problem domains it is not always possible to create complete, consistent models of the world thus, to obtain a meaningful conclusions, it is required not only to deal with what is possible, but also about what is probable [48]. On the other hand, graph theoretic side of graphical models helps describing knowledge of complex problems in simpler modules providing richer insights of domain in question. Through combined use of graphical structure and probabilities, Bayesian model provide capability of drawing conclusions on what information is known.

Consider a situation of a complex medical problem. Where information on patients in the form of symptoms, their test results, physical characteristics are given. Objective in this problem is to analyse given information to reach to a conclusion of possibility of disease. Given number of interconnected aspects like symptoms and other information for every patient, Bayesian network can help assist domain expert in diagnosis for the possibility of disease. Based on knowledge of how different entities are connected to each other, Bayesian network can reason to answer many interesting queries such as: possibility of presence of disease, given symptoms and few test results? Or given disease, what is the likeliness of positive test results?

In Bayesian network, graphical structure encode domain knowledge specified by setting interrelationship among random variables (like symptoms, physical characteristics of patients in example above) connected through an arrow from a source to a target variable in a system. The direct dependency between two variables using an arrow are often addressed as a *cause-effect* relationship in Bayesian terminology. Two components are customary to construction of BN namely, *qualitative* and *quantitative*. Qualitative component deals in identifying variables of interest for the domain, and then a graphical structure is formed by linking two variables if there exist a relational dependency between them. Bayesian network are kind of directed acyclic graphs (DAGs), meaning that edges in a graph have direction and, that there is no cycle within the graph. Quantitative component specify parameters indicating strength of relationship between connected variables in the Bayesian network.

Another most common kind of probabilistic graphical model is a Markov model [48]. Unlike Bayesian network, Markov model is undirected. However, both of these modesl consist of set of random variables representing our domain but, differ on how these random variables are connected. In a directed model, a directed edge is used to describe direct influence of one variable on another. Whereas, in a undirected model no such direction is defined. However, both these models describe similar perspective in terms of solving probabilistic queries. In Figures 3.2a and 3.2b a simple example of Bayesian and Markov model are presented respectively. The graphical structure of both these models consists of three nodes indicated by round boxes with their names appearing in corresponding boxes. In Bayesian network example there exist two direct dependencies: Y on (X and Z) and, (2) Z on X. In contrast to this, edges in Markov model correspond to a notion of direct probabilistic interaction between the neighbouring variables. For example, X depends on (Y and Z), Y on (X and Z) and Z on (X and Y). In this chapter, we focuss on theoretical concepts of Bayesian models since it form bases of this thesis.



(a) A Bayesian network (b) A Markov model

Figure 3.2: A simple Bayesian and Markov models

3.3 Summary of Notations

Table 3.1 summarizes notation used for Bayesian networks in this chapter.

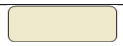
Notation	Description
	Discrete variable
	Continuous variable
BN	Bayesian network
CPT	Conditional probability table
CPD	Conditional probability distribution
X, Y, Z	Variables/nodes
X, Y, Z	Set of variables/nodes
C	Child node
$X_{\diamond}$	Set of discrete nodes in a model
X_{τ}	Set of continuous nodes in a model
$\lvert\mathbf{X}\rvert$	Total number of variables in a model
x, y, z, x_i, y_i, c_i	States of variables
Val(X)	Set of configuration/states of variable X
$\lvert$Val(X)$\rvert$	Total number of states X takes
Anc(X)	Ancestors of X in model
Dec(X)	Descendants of X in model
NonDes(X)	Non descendants of X in model
G	Bayesian graph
Pa(C)	Parents of child node C in a model
E	Set of edges in a model
X_p	Set of parent nodes in a model
P(X = x_i)	Probability of X in x_i
P(C = c_i $\mid$ Pa(C))	Conditional probability of child node C in c_i given parents
ε	Evidence in a model
q	Query node in a model
$X \rightarrow Y \rightarrow Z$	Path in a model
$X \rightleftharpoons Y \rightleftharpoons Z$	Trail in a model

Table 3.1: Bayesian network: notations and basic concepts

3.4 Understanding Bayesian network Model

3.4.1 Variables, Nodes and States

A variable X in BN can be defined as a concept about which information can be stored. The notion of variables and nodes are often used interchangeably in Bayesian terminology. Common types of nodes used in Bayesian network are: *discrete* and *continuous*. Each discrete node in BN represents an exhaustive set of mutually exclusive events, often referred to as states or values. Mutually exclusive and exhaustive property means that the variable must take on exactly one of the possible value. In contrast, continuous variable has infinite number of states. In Table 3.2 few examples of variables are presented.

Node name	Type	Values/States
Gender	Discrete	male, female
Height	Discrete	short, medium, tall
Weather	Discrete	sunny, cloudy
Temperature	Continuous	-∞,∞

Table 3.2: Examples of variables/nodes in Bayesian network

We define set of states a variable X can take by Val(X). Total number of states of a variable X is denoted by |Val(X)|. For example, variable Gender defined in Table 3.2 has |Val(Gender)| = 2. The notation X = x, denote the fact that variable X attains the value x. We define probability of variable X in some state x_i by P(X = x_i). For convenience we may refer P(X = x_i) as P(X = x).

3.4.2 Taxonomy on Bayesian Networks

A Bayesian graph is a structure ***G*** consisting of set of **nodes/vertices** (*X*) and **edges/arcs** (*E*). A node is a **parent** of a **child**, if there is an arc from the former to the latter. For example in relational dependency X → Y, X is a parent whereas, Y is its child. In very simple terms, parent nodes in Bayesian network accounts for trends in application domain, and nodes which are influenced by these trends are child nodes. We use the notation Pa(C) to represent set of its parent nodes for any child node C.

In a Bayesian graph **G** = (*X*, *E*), set of variables $X = \{X_1, X_2 ...., X_n\}$ are said to form a **path** if, for every *i = 1,..., n-1*, we have $X_i \rightarrow X_{i+1}$. A **trail** in **G** is also collection of

edges which is like a path, but unlike path trail may have edges in any direction. More formally, set of variables $X = \{X_1, X_2...., X_n\}$ forms a trail in **G** if, for every $i = 1,...,$ $n-1$, we have $X_i \rightleftharpoons X_{i+1}$.

In a directed chain of nodes, one node is a **ancestor** of another if it appears earlier in the chain. Whereas, a node is a **descendent** of another node if it comes later in the chain. Since BNs are directed acyclic graphs where loops are disallowed so, no child node can be its own ancestor or descendent. We use the convention of representing set of ancestor and descendant for node X as: **Anc(X)** and **Dec(X)** respectively.

Once graphical topology of the network is specified, next step is to quantify relationships. Quantification of relationships refers to mechanism of describing degree of correlation that exists between two connected nodes in the graph . As discussed in Section 3.4.1, variables in Bayesian network can be either discrete or continuous so, they are treated in a different way in defining degree of correlation. In a discrete framework, theory of probability is used whereas; concept of density functions using Gaussian distribution is used for continuous nodes in the model.

For each discrete node in the model, conditional probability distribution is specified which takes the form of **conditional probability table** (CPT). An unconditional node, i.e., a node which is not child node in the network, entries in CPT is like a prior or unconditional probabilities defining plausibility of being in a specified state. Equation 3.17 defines CPT entries for an unconditional node X in the Bayesian graph subject to condition that $\sum_{i=1}^{|Val(X)|} P(X = x_i) = 1$.

$$P(X) = (P(X = x_1), P(X = x_2), ..., P(X = x_{|Val(X)|}) \tag{3.17}$$

On the other hand, for every child/conditional node in BN, first all possible combinations of values of parent nodes are specified. Then, for each combination which is also called as *instantiation*, probability in each distinct value in child node is specified. For a conditional probability we use to notation $P(Y \mid X)$ where, Y is conditioned on X. For CPT of a child node, we have a probability distribution $P(Y = y \mid X = x)$ for each combination of $y \in Val(Y)$ and $x \in Val(X)$. Unlike parametrization in the discrete framework, possible space in the continuous case is not bounded. In this case, multivariate Gaussian is used to define parameters for interrelated variables. We discuss how continuous variables are modelled in Section 3.6 of this chapter. For now we assume, Bayesian network encodes only discrete variables.

Definition: A discrete Bayesian network **G** = (X, E) consists of:

1. A DAG **G** = (X, E) with set of nodes X and directed links E.
2. A set of conditional probability distribution, P, containing one distribution, P(X | Pa(X)), for each random variable X $\in$ X.

3.4.3 Bayesian Network: Example

In Figure 3.3, we present a simple Bayesian network relating to a scenario of potential fire diagnoses in a building taken from Netica network repository [23]. This Bayesian network has six boolean variables as indicated by six circles with name of the variables written within respective circles. The five directed arrows in the model reveals relational dependency between set of variables disseminating knowledge on how system works. This simple model explains the fact that alarm (Alarm or A in BN) in the building can be caused by the two factors namely, fire (Fire or F in BN) and tampering (Tampering or Ta in BN). That is, alarm in the building can be result of actual fire or when someone intentionally plays with it. This fact is encoded by directed arrows starting from variables Fire and Tampering and ending on Alarm. Fire in the building can give rise to smoke, indicated by a directed arrow between fire (Fire or Fi in BN) and smoke (Smoke or S in BN). The status of alarm may cause people staying in the building leaving (Leaving or L in BN) their houses, directed arrow between these nodes represent this fact. A report (Report or R in BN) is maintained on people leaving the house, a direct dependency between leaving and report reveals this fact. For simplicity, we restrict naming variables by their initials indicated in braces next to their names in the model.

The BN encode several parent and child nodes. For example variables: Fi and Ta are parents of A, and A is child of both Fi and Ta. In other words, Fi and Ta are "causes" and, A is their "effect". Ancestor nodes for L, i.e., Ans(L) = Fi, Ta, A, S whereas, example of descendant of L represented by notation Dec(L) is R. Two examples of directed path in this BN are: (1) Fi $\rightarrow$ A $\rightarrow$ L $\rightarrow$ R and, (2) Fi $\rightarrow$ S. Two examples of trails are: (1) S $\leftarrow$ Fi $\rightarrow$ A $\leftarrow$ Ta and, (2) S $\leftarrow$ Fi $\rightarrow$ A $\rightarrow$ L $\rightarrow$ R.

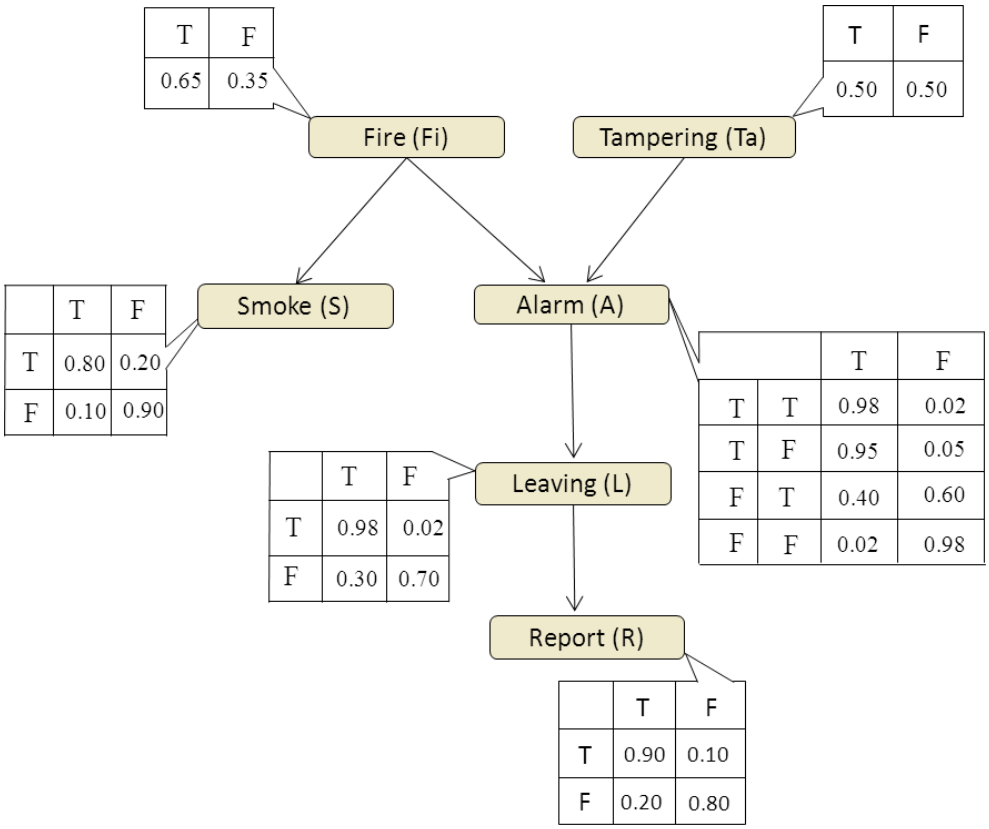


Figure 3.3: Bayesian network on a fire diagnose problem

As opposed to qualitative component of this BN discussed above, quantitative information is represented by a set of an unconditional and conditional probabilities associated with each node. Let each node is binary with possible states as True (T) and False (F). Since Fi and Ta do not have any parent so, their respective table contains unconditional probabilities stating chances of these variables in states T or F. For example, probability of fire is 65% and chances of not seeing the fire is 35%. In contrast to unconditional nodes, conditional node S is associated with the conditional probability table indicating probability of smoke in presence and absence of fire. For example, fire in the building causes smoke to rise in the building with 80% probability. Similar information is encoded in CPTs of conditional variables A, L and R.

3.4.4 Independence in Bayesian Networks

Bayesian networks provide a very intuitive language for representing dependence and independence statements among problem-domain variables. An arc joining two variables in the BN represent direct dependency whereas, lack of arc represent conditional independence assumptions [48; 56]. This Section is dedicated in describing independence properties in BN and how they are useful.

Bayesian theory revolves around key concepts of dependency, independency and conditionally independence among variables. In very general terms, two events are said to be dependent if knowledge of one provides predictive value for of another. For example, consider a simple BN in Figure 3.4 where, Rain is dependent on Sun. Knowing its sunning or not, we can predict possibility of rain. However there are situations when knowledge of one variable provides no predictive value for the knowledge of another. For example, knowledge on number of people on the street provides no prediction on probability of rain. More formally, independent events can be described using definition below.

Definition: A variable X is independent of another variable Y with respect to a probability distribution P if Equation 3.18 holds.

$$P(X|Y) = P(X), \forall x_i \in Val(X), \forall y_j \in Val(Y) \tag{3.18}$$

Conditional independence comes into play when we have multiple variables that can all be correlated. Two events are said to be conditional independent when observations on an additional event is given. Consider BN in Figure 3.5 which is extension of one presented in Figure 3.4. In this case, while it is true that knowledge of sun provides predictive value for carrying umbrella (variable Umbrella in BN) because no sun means it is more likely to be raining, and thus more likely to carry an umbrella. Interestingly enough, this predictive value is entirely mediated through Rain. If we already know status of rain, knowing its sunny or not does not help further predict status of carrying umbrella. Here, the two variables Umbrella and Sun are conditionally independent given knowledge of Rain.

The concept of conditional independence is central to the power of Bayesian networks. Conditional independence assumptions encoded in the Bayesian structure provides several benefits: (1) conditional independence property in BN provides key to Bayesian inference through which complex probabilistic queries can be answered (2) by exploiting conditional independence statements in BN, computational cost involve in inference can be substantially reduced and, (3) this property can easily be inferred through Bayesian graphical structure which can help users with key insights of domain knowledge.

We describe formal definition of conditional independence in BN below [48].

Definition: Let a Bayesian network structure **G** consist of set of random variables $X = \{X_1, X_2...., X_n\}$. Let $Pa(X_i)$ denote the parents of X_i in G, and $NonDes(X_i)$, denote the variables in the graph that are not descendants of X_i. Then **G** encodes the following set of conditional independence assumptions [48].

$$For\ \ each\ \ X_i \in X : (X_i \perp NonDes(X) \mid Pa(X_i)) \tag{3.19}$$

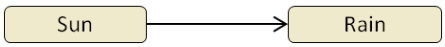


Figure 3.4: Bayesian network showing relational dependency between variables Sun and Rain

Figure 3.5: Bayesian network showing variable Umbrella is conditionally independent of Sun, given Rain

3.4.5 Joint probability distribution in Bayesian networks

The chain rule in probability theory allows us to factorize joint probabilities as represented by Equation 3.20. However, graphical structure of BN gives advantage of representing joint probability distributions concisely and compactly. Consider a Bayesian network containing n nodes represented as X_1,....., X_n. Recalling from Equation 3.19 that the structure of a BN implies the value of a particular node is conditioned only on

the values of its parent nodes, this reduces Equation 3.20 to Equation 3.21.

$$P(X_1 = \ x_1, X_2 = x_2, ..., X_n = x_n) = P(X_1 = x_1) \times P(X_2 = x_2 | X_1 = x_1) \times \\ ... \times \ (X_n = | X_1 = x_1, ..., X_{n-1} = x_{n-1}) \tag{3.20}$$

$$P(X_1 = x_1, X_2 = x_2, ..., X_n) = \prod_{i=1}^{n} P(X_i = x_i \mid Pa(X_i)) \tag{3.21}$$

In general, joint distribution of a set of variables is an exponentially-sized object. If all the variables are binary, the joint over n variables has 2^n parameters. For example, on ten binary variables using Equation 3.20 may require 2^{10} = 1024 storage space. On the other hand, memory space of ($2^p \times |\mathbf{X}|$) where, p implies maximum number of parent nodes a variable in BN has is required in case of Bayesian networks. That is, if maximum of three parent nodes are restricted in BN of total ten nodes then, ($2^3 \times 10$) = 80 values are required.

We now present example of joint probability calculation using BN in Figure 3.3. Suppose we are interested in computation of situation, (Fi = T, Ta = F, S = T, A = T, L = T, R = T). Equation 3.22 below explains how the said joint probability is calculated using Equation 3.21. The computation yields probability of 16.7%.

$$\begin{aligned} P(Fi = T, Ta = F, S = T, A = T, L = T, R = T) = \ & P(R = T | L = T) \times P(L = T | A = T) \\ & \times \ P(A = T | Fi = T, Ta = F) \\ & \times \ P(S = T | Fi = T) \ \times \ P(Fi = T) \\ & \times \ P(Ta = F) = 16.7\% \end{aligned} \tag{3.22}$$

3.5 Flow of Information in Bayesian Networks

A key task in Bayesian network is the computation of new beliefs when new information is available. In particular, observations which are known are said to be ***evidence*** in Bayesian terminology. Let evidence ε, is the information received from external sources about the possible states of the subset of the variables of the network. In presence of ε, posterior probabilities of the form P(Y $\mid$ ε) are computed. We follow the notation of attaching the label ε to variables in the Bayesian network which are known observations.

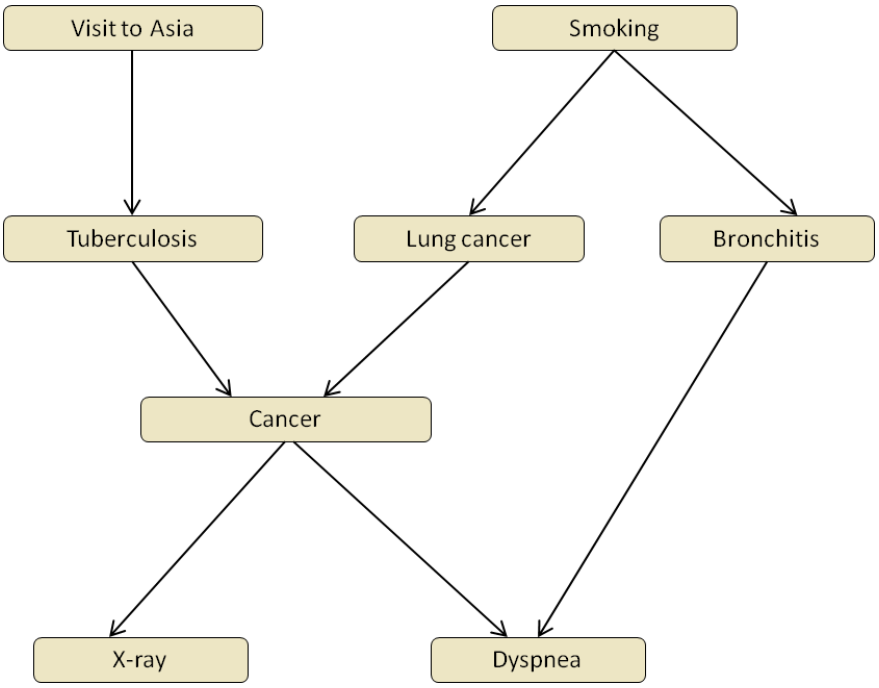


Figure 3.6: Bayesian network on a medical problem

Based on what evidence is given, there exist three types of connections through which information flows in BN [44; 48]. We now present analysis on each these connections using small fictitious BN represented in Figure 3.6 on a medical problem.

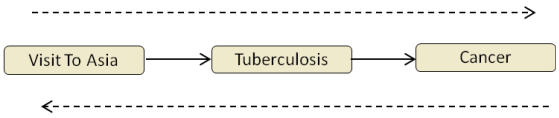
Example: 2.1 Medical knowledge encoded in the BN represented in Figure 3.6 states that people who recently visited Asia (Visit to Asia in BN) are more likely to suffer from tuberculosis (Tuberculosis in BN) while, smoking (Smoking in BN) is a single risk factor known for both lung cancer (Lung cancer in BN) and bronchitis (Bronchitis in BN). Patient can have positive or negative x-ray (X-ray in BN) depending on whether he suffers from tuberculosis or lung cancer (Tuberculosis or Lung cancer in BN). That is, result of x-ray report does not discriminate between lung cancer and tuberculosis. Shortness of breath which is called as Dyspnea (Dyspnea in BN) in medical terms could be caused if the person is suffering either from bronchitis (Bronchitis in BN) or in the presence of tuberculosis/lung cancer.

1. Serial connections: These connections are often addressed as causal chains. In such connections, two nodes are directly connected. Consider a subgraph of BN presented in Figure 3.6 on a medical diagnose problem in Figure 3.7a. A three node

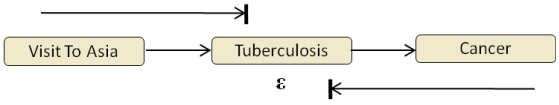
network shows nodes, Visit to Asia and Cancer are not directly connected, but indirectly connected through a trail between them via node, Tuberculosis. Serial connections like presented in Figure 3.7a also implies notion of indirect interactions in BN. In such connection, if no evidence is available, information can travel between the nodes in either direction as shown using dashed arrows in Figure 3.7a. However, when a state of the middle variable is known for sure, then flow of information between the other two variables cannot take place through this connection, refer Figure 3.7b. Given knowledge on middle node, Tuberculosis blocks the flow of information over a serial connection. This suggest variable, Visit to Asia cannot influence variable, Cancer when evidence on variable, Tuberculosis is given.

2. Diverging connections: Example of diversing connection from BN in Figure 3.6 is presented in Figure 3.7c. As indicated, such connections have a common cause. Like serial connections, information between nodes can flow in absence of evidence. Information channel between nodes Lung cancer and Bronchitis gets blocked with evidence on variable Smoking, refer Figure 3.7d. Thus conclusion here is identical to the previous connection: Lung cancer can influence Bronchitis via Smoking if and only if Smoking is not observed.

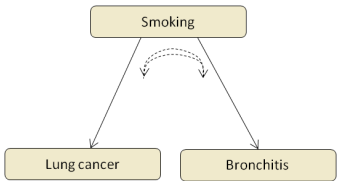
3. Converging connections: Contrary to serial and diverging connections, a converging connection does not transmit information between nodes if no evidence is available for the middle node. Such connections have a common child node for more than one parent node. They are often addressed as **v-structure** [48]. Example of converging connection from BN in Figure 3.6 is shown in Figure 3.7e where it is shown using dashed arrows that Dyspnea blocks information passage between nodes Cancer and Bronchitis. In other words, if no evidence is available about the state of Dyspnea then information about the state of Bronchitis will not provide any derived information about the state of Cancer. It explains the fact that Cancer is not an indicator of Bronchitis. However, same channel is unblocked with known observation on variable Dyspnea. This fact is illustrated in Figure 3.7f. Given observation on a state of Dyspnea, information about the state of Bronchitis will provide an explanation for the evidence that was received about the state of Dyspnea. The opposite also holds true.



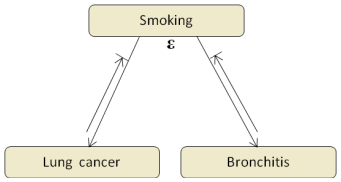
(a) A serial connection in absence of evidence



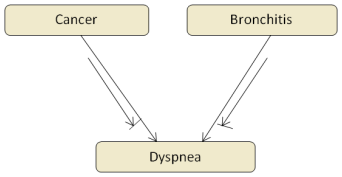
(b) A serial connection in presence of evidence



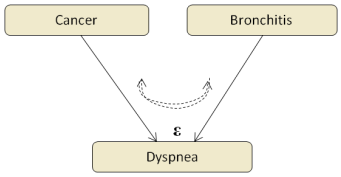
(c) A diverging connection in absence of evidence



(d) A diverging connection in presence of evidence



(e) A converging connection in absence of evidence



(f) A converging connection in presence of evidence

Figure 3.7: Information flow in Bayesian networks

3.5.1 D-separation

Based on three types of connections discussed above, information flow in BN can be summarized using definition below [48]

Definition: Let **G** be a BN structure, and $X_1 \rightleftharpoons ... \rightleftharpoons X_n$, a trail in **G**. Let Z be a subset of observed variables. The trail $X_1 \rightleftharpoons ... \rightleftharpoons X_n$ is active given Z if

- Whenever we have a v-structure $X_{i-1} \rightarrow X_i \rightarrow X_{i+1}$, then V_i or none of its descendants are in Z
- No other node along the trail is in Z

Given Bayesian graphical structure, there could be more than one trail between two nodes. Like for an example BN in Figure 3.6, there are two trails between nodes, Smoking and X-ray: (1) Smoking $\rightarrow$ Lung cancer $\rightarrow$ Cancer $\rightarrow$ X-ray and, (2) Smoking

$\rightarrow$ Bronchitis $\rightarrow$ Dyspnea $\leftarrow$ Cancer $\rightarrow$ X-ray. For any number of trials connecting two nodes, one node can influence another if any trail consists of connections such that information between them can flow. Intuitively, these connections joining variables in BN help comprising general rules for reading relevant and irrelevant relations for two variables, given set of other variables.

The concept of D-separation provides notion of interpreting conditional dependence and independence properties in BN and is due to [57]. In other words, the D-separation criteria can be used to answer queries of the kind "are X and Y independent given Z"? or queries of the kind "is information about X irrelevant for our belief in Y given information on Z"?. The formal definition of D-separation from [48] is defined below.

Definition: Let X, Y, Z be three sets of nodes in G. We say that X and Y are D-separated given Z, denoted by d-sep$_G(X; Y \mid Z)$, if there is no active trail between any node X $\in X$ and Y $\in Y$ given Z.

Following are few examples of d-separation from Bayesian network in Figure 3.6:

1. d-sep$_G$(Smoking; Dyspnea | Bronchitis, Cancer): Observation on Bronchitis blocks trail Smoking $\rightarrow$ Bronchitis $\rightarrow$ Dyspnea and, evidence on Cancer blocks the trail Smoking $\rightarrow$ Lung cancer $\rightarrow$ Cancer $\rightarrow$ Dyspnea

2. d-sep$_G$(Tuberculosis; Bronchitis | Cancer, Smoking): The trail Tuberculosis $\rightarrow$ Cancer $\rightarrow$ Dyspnea $\leftarrow$ Bronchitis gets blocked by Cancer whereas, Smoking blocks the chain Tuberculosis $\rightarrow$ Cancer $\leftarrow$ Lung cancer $\leftarrow$ Smoking $\rightarrow$ Bronchitis $\rightarrow$ Dyspnea

3.6 Continuous Variables and Bayesian Networks

In this Section, we describe how continuous variables can be integrated into the Bayesian network framework. The most commonly used distribution of representing continuous variables in Bayesian framework is ***Gaussian***. Multivariate Gaussian distributions form the basis of describing continuous variables in the model. We first start with discussion on multivariate Gaussian distributions. Then, we describe two different kinds of Bayesian networks which involve continuous variables namely, ***Gaussian Bayesian networks*** and ***Hybrid Bayesian networks***. Gaussian Bayesian networks entail a pure

continuous case, where all variables are continuous in nature, both as parents and as children. On the other hand, Hybrid Bayesian networks involve both discrete and continuous variables.

Density function for a multivariate Gaussian distribution over X_1,...,X_n is characterized by an n-dimensional mean vector μ, and a symmetric $n \times n$ covariance matrix Σ defined in Equation 3.23. Where, $|\Sigma|$ represents determinant of Σ. In Figure 3.8 and Figure 3.9 shows two multivariate Gaussians, one where variables are independent, and one where they are dependent respectively.

$$p(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \tag{3.23}$$

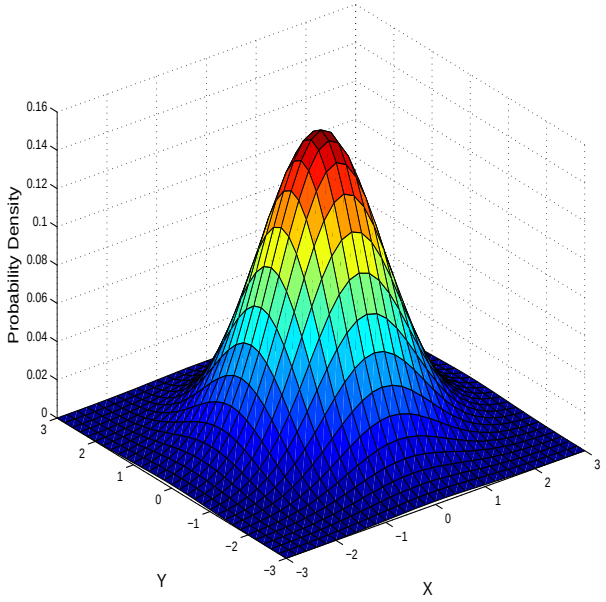


Figure 3.8: Gaussian over two independent variables

The joint normal distribution over X and Y where $X \in \Re^n$ and $Y \in \Re^m$ is defined in Equation 3.24 [48]. Where $\mu_X \in \Re^n$, $\mu_Y \in \Re^m$, Σ_{XX} is a matrix of size $n \times n$, Σ_{XY} is a matrix of size $n \times m$, $\Sigma_{XY} = \Sigma_{YX}^T$ is a matrix of size $n \times m$ and Σ_{YY} is a matrix of size m

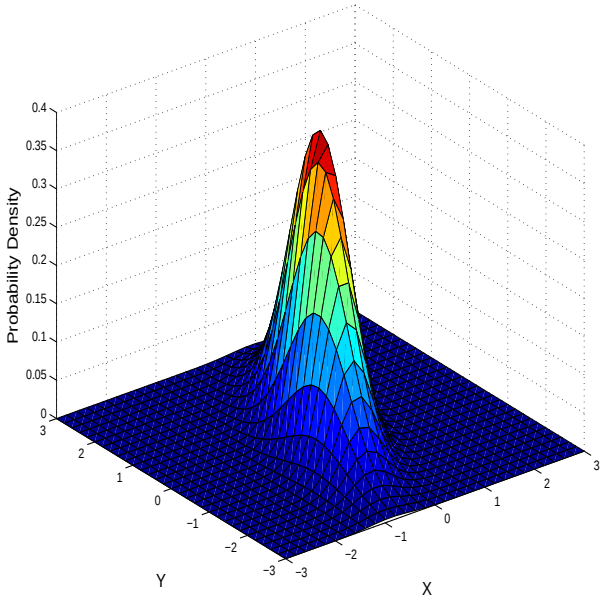


Figure 3.9: Gaussian over two dependent variables

$\times m$.

$$p(X,Y) = N\left(\begin{pmatrix}\mu_X \\ \mu_Y\end{pmatrix}; \begin{pmatrix}\Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY}\end{pmatrix}\right) \tag{3.24}$$

Given joint normal distribution represented by Equation 3.24, marginal distribution over some subset of the variables Y can be computed directly from entries in mean and covariance matrix defined in the equation [48]. Consider joint normal distribution of two variables X and Y defined in Equation 3.25. Then, variable Y is normally distributed with $\mu = 3$ and $\Sigma = 5$

$$p(X,Y) = N\left(\begin{pmatrix}1 \\ 3\end{pmatrix}; \begin{pmatrix}4 & 2 \\ 2 & 5\end{pmatrix}\right) \tag{3.25}$$

3.6.1 Gaussian Bayesian Networks

A Gaussian Bayesian network is a kind of Bayesian network all of whose variables are continuous, and where all of the CPDs are *linear Gaussians* [48; 56; 38; 44]. The term

linear defines that if a continuous variable has one or more continuous variables as parents, the mean may depend linearly on the state of the continuous parent variables. Continuous parent variables of discrete variables are disallowed in this framework. More formally, let Y be a continuous variable with continuous parents X_1, X_2,...,X_n. Then, Y has a linear Gaussian model if there are parameters $\beta_0, \beta_1, ..., \beta_n$ and σ^2 such that [48]

$$P(Y \mid X_1, X_2..., X_n) = N(\beta_0 + \beta_1 X_1 + ... + \beta_k x_k; \sigma^2) \tag{3.26}$$

In vector notation, Equation 3.26 can be rewritten as Equation 3.27.

$$P(Y \mid X) = N(\beta_0 + \beta^T X; \sigma^2) \tag{3.27}$$

The distribution of Y is a normal distribution, p(Y) = N (μ_Y; σ_Y^2) where μ_Y and σ_Y^2 is defined as follows.

$$\mu_Y = \beta_0 + \beta^T \mu \tag{3.28}$$

$$\sigma_Y^2 = \sigma^2 + \beta^T \Sigma \beta \tag{3.29}$$

Consider Bayesian network in Figure 3.10. Where associated with each independent node X and Y normal distribution is defined whereas for dependent node Z, normal distribution is defined using Equation 3.26. Using Equations 3.28 and 3.29 mean and standard deviation on node Z is computed as below.

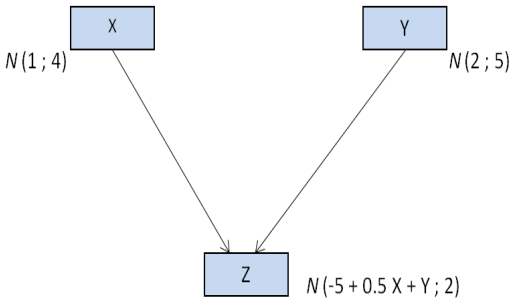


Figure 3.10: A Gaussian Bayesian network

$$\mu_Z = -5 + 0.5 \times 1 + 1 \times 2 = -2.5 \tag{3.30}$$

$$\Sigma_Z = 2 + (0.5)^2 \times 4 + 1^2 \times 5 = 8 \tag{3.31}$$

3.6.2 Hybrid Bayesian Networks

Hybrid Bayesian networks incorporate both discrete and continuous variables. For clarity, let variable set X is partitioned into the set of continuous variables, X_τ and, the set of discrete variables, $X_\diamond$. In hybrid Bayesian networks there exist conditional probability distribution for each variable in $X_\diamond$ and one density function for each continuous random variable in X_τ. We restrict discussion on hybrid Bayesian networks where there exists condition on its graphical structure that any discrete variable in network cannot have continuous parents. Hybrid network with this specific condition is called ***linear conditional Gaussian networks*** (LCG for short) [48; 44]. In an LCG networks each node is either a discrete random variable with a finite state of mutually exclusive and exhaustive states or a continuous random variable with a linear conditional Gaussian distribution conditional on the configuration of its discrete parent variables. If a continuous variable has one or more continuous variables as parents, the mean may depend linearly on the state of the continuous parent variables. More formally, a continuous random variable X has a linear conditional Gaussian distribution conditional on the configuration of parents variables ($Z \subseteq X_\tau$, $I \subseteq X_\diamond$) if Equation 3.32 holds.

$$P(X \mid Z = z, I = i) = N(A(i) + B(i)^T z, C(i)) \tag{3.32}$$

In Equation 3.32, A stands for table of mean values (one value for each configuration i of the discrete parent variable I), B is a table of regression coefficient vectors (one vector for each configuration i of I with one regression coefficient for each continuous parent variable) and, C is the table of variances (one for each configuration i of I). Equation 3.32 also states that mean at X depends linearly on the values of the continuous parent variables Z, while the variance is independent of Z. The quantitative part of an LCG Bayesian network consists of a conditional probability for each $X \in X_\diamond$ and a conditional Gaussian distribution for each $X \in X_\tau$. For each $X \in X_\tau$ with discrete parents, I, and continuous parents, Z, one dimensional Gaussian probability distribution for each configuration of i of I is specified.

Definition: An LCG Bayesian network **G** = (X, E, P, F) consist of:

- A DAG **G** = (X, E) with sets of variables X and directed edges E.
- A set of conditional probability distribution, P, containing one distribution, P(X |

Pa(X)), for each discrete random variable X.

- A set of conditional linear Gaussian probability density functions, F, containing one density function, P(W | Pa(W)), for each continuous random variable W.

In Figure 3.11, a hypothetical Hybrid Bayesian network encoding a general knowledge on persons choice of employment (Employment in BN), income (Income in BN), expenditure (Expenditure in BN) he makes, lifestyle (Lifestyle) he follows and mortgage (Mortgage in BN) he can avail from bank. We consider variables, Income, Expenditure and Mortgage as of continuous nature whereas; Employment and Lifestyle are taken as discrete. In order to distinguish discrete variables from continuous variables in the model, we use rounded box for representing discrete variables while, square box for continuous variables. Associated with variable Employment are the unconditional probabilities in two of its mutually exhaustive states namely, business and private. For variable Lifestyle, the conditional probability distribution in its states (high and low), given configuration of parent variable Employment is defined by the table associated with it. On the other hand, the parametric information for variable Income is represented with a pair (μ; σ^2) where, μ represents the mean and σ^2 the variance. For variable Expenditure, parametric information is represented using Equation 3.32, i.e., for each exclusive state of employment, probability density function is specified where, the mean depends on the parent node, Income. Similar information is enclosed for variable Mortgage.

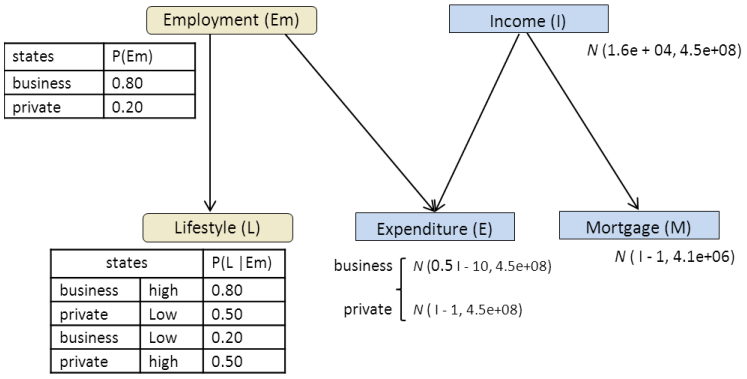


Figure 3.11: A Hybrid Bayesian network

3.7 Inference in Bayesian Networks

They key property of Bayesian network is that it provides separation of *knowledge* about domain and *reasoning* [48]. That is, once Bayesian network is encoded with domain knowledge, task such as reasoning and decision making can be employed without updating the Bayesian model. Reasoning in BN is done through *inference* which refers to the task of solving probabilistic queries based on relationships encoded in the model and evidence known about the situation at hand. In particular, on applying evidence about observation, a Bayesian mathematical mechanics updates the probabilities of all the other variables that are connected to the variable representing the new evidence. The updated probabilities reflect the new levels of belief in all possible outcomes coded in the model. The beliefs originally encoded in the model are known as *prior probabilities*, because they are entered before any evidence is known about the situation. The beliefs computed after evidence is entered are known as *posterior probabilities*, because they reflect the levels of belief computed in light of the new evidence. This process is also known as *probability propagation* or *belief updating* in Bayesian terminology. The key to Belief updating in BN is *"information flow mechanism"* discussed in Section 3.5. Bayesian inference is not limited to the directions of the arcs in the model, i.e., we can reason either in top to bottom fashion or from bottom to top.

One particular type of probabilistic inference task in BN is the task of computing the posterior marginal of an unobserved variable Y when there is no evidence available, i.e., $\varepsilon = \emptyset$. In a BN over n discrete variables $X = \{X_1, X_2...,X_n\}$, if we are to compute marginal on any variable Y then, it can be computed by exploiting the chain rule in BN (refer Equation 3.21). Equation 3.33 defines how P(**Y**) is calculated. Where notation $\setminus$ stands for exclusion.

$$P(Y) = \sum_{X_i \in X \setminus Y} \prod P(X_i \mid Pa(X_i) \tag{3.33}$$

Besides computing marginal distribution, inference in BN can be used for many interesting reasoning. In this section, we discuss different types of reasoning supported in BN.

3.7.1 Reasoning in Bayesian networks

Bayesian networks support three types of reasoning: ***causal***, ***diagnostic*** and ***inter-causal*** [48]. The main idea behind inference mechanism involved in these reasonings

is, generation of joint distribution and exhaustively summing out observations nodes, i.e., variables for which evidence are given and a query node which actually is the interest node of which posterior is to be calculated. We follow notation of representing query node as **q**. Below we discuss on each of the reasoning type with an example illustration using Bayesian network on a fire diagnose problem taken previously.

1. Causal: It is reasoning about new information about *causes* to new beliefs in *effects*, following the direction of arrows in the network. Given knowledge of how system works in case of fire in the form of Bayesian network represented in Figure 3.3, suppose we want to know probability of people leaving the building in the presence of fire, i.e., P(L = T | Fi = T). Figure 3.12a explains the scenario of causal reasoning for the said query. Here node, L is a query node indicated by a dashed outline in the figure. The conditional probability, P(L = T | Fi = T) is solved using Equation 3.34. Using joint probability in this BN as presented in Equation 3.22, and concept of marginalization, Equation 3.34 can be rewritten as Equation 3.35. Solving Equation 3.35 using CPTs associated with variables in the network results in probability of 96.5%

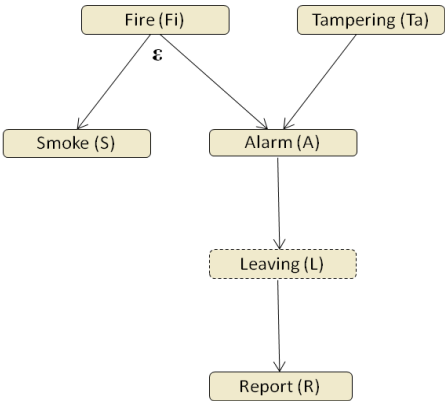
$$P(L = T|Fi = T) = \frac{P(L = T, Fi = T)}{P(Fi = T)} \tag{3.34}$$

$$P(L = T \mid Fi = T) = \frac{\sum_S \sum_{Ta} \sum_A \sum_R P(L = T, Fi = T, A, S, R, Ta)}{P(Fi = T)} = 96.5\% \tag{3.35}$$

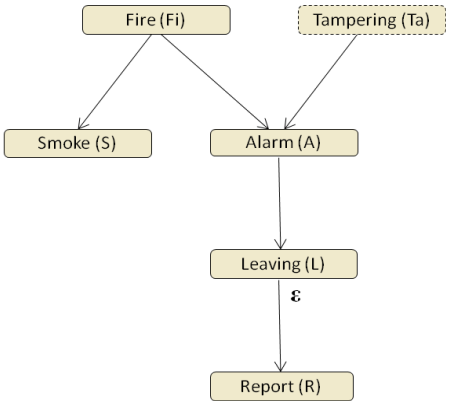
2. Diagnostic: It operates from *effects* to a *cause*, i.e., in the reverse direction of arrow. Consider a situation where people have left their houses because the alarm of the building went on. And now, we want to know probability that alarm went on because of tampering, i.e., P(Ta = T | L = T). BN in Figure 3.12b explains this scenario. Where L is the evident node whereas, T is the query node. Using Equation 3.36 the interested query is solved.

$$P(Ta = T \mid L = T) = \frac{\sum_S \sum_{Fi} \sum_A \sum_R P(Ta = T, L = T, S, Fi, A, R)}{P(L = T)} = 53.3\% \tag{3.36}$$

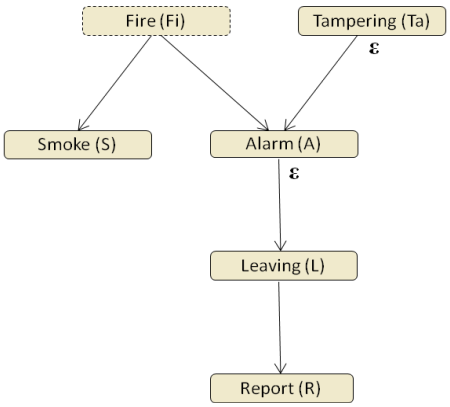
3. Intercausal This form of reasoning involves about the mutual *causes* of a common *effect*, i.e., those which involves v-structure (refer Section 3.5) in BN. It is also known as ***explaining away***. Suppose we learn that alarm went on. This will give rise to our probability in both of its cause, i.e., fire and tampering. Suppose later we discover that alarm was caused because of tampering. This new information explains the observed status of alarm, which in turn lowers our probability in presence of fire. So, even though the two causes were initially independent, with knowledge of the effect the presence of one explanatory cause renders an alternative cause less likely. In other words, the alternative cause has been *explained away*. The situation is explained in Figure 3.12c.



(a) Example of Causal reasoning in Bayesian networks



(b) Example of Diagnostic reasoning in Bayesian networks



(c) Example of Intercausal reasoning in Bayesian networks

Figure 3.12: Reasonings in Bayesian networks

3.7.2 Variable Elimination in Discrete Bayesian Networks

The problem of inference in Bayesian networks is NP-hard and therefore it requires exponential time in the worst case. However, in practice for real-world problems, Bayesian inference can be tackled efficiently using *exact* and *approximate* inference algorithms [48]. We now discuss an efficient *variable elimination* approach for Bayesian inference.

We show using a very simple Bayesian network example, how a general probabilistic query of computing marginal distribution is exponential in nature, refer Figure 3.13. Suppose we are to compute P(K). In order to compute this, we have to calculate the joint probability and sum out all variables leaving K. More precisely, Equation 3.37 solves marginal on K.

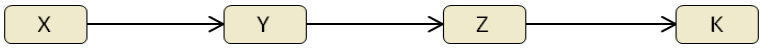


Figure 3.13: A simple Bayesian network

$$P(K) = \sum_X \sum_Y \sum_Z P(X,Y,Z,K) = \sum_X \sum_Y \sum_Z P(X)P(Y|X)P(Z|Y)P(K|Z) \tag{3.37}$$

If each variable takes m values then, Equation 3.37 generates m^4 probabilities in the joint distribution that are summed over. So in general, for n nodes in BN taking m values, we may require m^n probabilities in order to compute marginal. Variable elimination approach for Bayesian approach fundamentally contains two main ideas that help address exponential blowup of the joint probabilities. And, these are:

- This approach exploits basic arithmetic properties. For example, expression, (aA_1 + aA_2 + aA_3 +aA_4) can be rewritten as, a(A_1 + A_2 + A_3 +A_4) which reduces the problem of four multiplications to one.

- By taking advantage of Bayesian graphical structure, many sub expressions in the joint only depend on a small number of variables than on total number of variables present in the network.

- While computation, intermediate results are cached in order to avoid recomputation.

Consider again Bayesian network in Figure 3.13 and problem of computing P(K). Using basic arithmetic and the structure of the network, we can rearrange Equation 3.37 using subexpression below.

$$P(Y) = \sum_{X} P(X)P(Y|X) \tag{3.38}$$

$$P(Z) = \sum_{Y} P(Z)P(Z|Y) \tag{3.39}$$

$$P(K) = \sum_{Z} P(Z)P(K|Z) \tag{3.40}$$

Putting above expressions back in Equation 3.37 we get Equation 3.41 below:

$$P(K) = \sum_{Z} P(K|Z) \left(\sum_{Y} P(Z|Y) \left(\sum_{X} P(X)P(Y|X) \right) \right) \tag{3.41}$$

The inner expression (or P(Y)) is computed first for all values of Y and stored so that they are computed once. Then, P(Z) is computed with the values of P(Y) and stored. Finally, P(K) is computed with the stored values of P(Y).

In general terms, for a BN that has a structure of chain with n variables $X_1 \rightarrow X_2 \rightarrow ... \rightarrow X_n$, where each variable has m possible values, computing $P(X_i{+}1)$ can be defined recursively as in Equation 3.42. Each recursive step is $O(m^2)$ and recursing through all n variables in network yields computation of $O(nm^2)$ operations in the worst case [48]. This is much smaller than generating the full m^n probabilities to sum over in the joint distribution.

$$P(X_{i+1}) = \sum_{X_i} P(X_{i+1}|X_i)P(X_i) \tag{3.42}$$

The basic idea in the algorithm is summing out variables one at a time. When any variable is summed out, multiplication is performed on all the factors that mention that variable, generating a product factor. Now, variable is summed out from the combined factor, generating a new factor to deal with. Algorithm of variable elimination can be found in [48].

We now present an example illustration on variable elimination algorithm using Bayesian network presented in Figure 3.14. Suppose we are interested in probabilistic query, P(U | X, Y). In order to solve this problem, we need to marginalise over W and

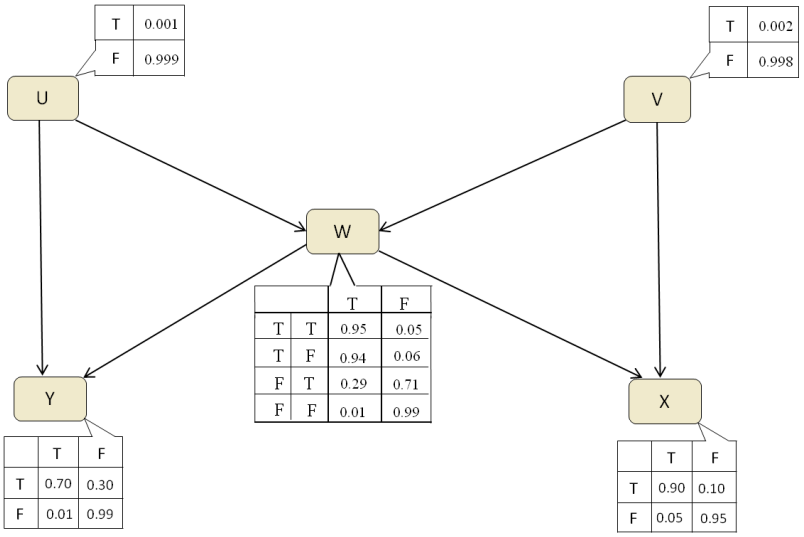


Figure 3.14: A hypothetical Bayesian network

V. We are given states of variables X and Y while, U is the query node. Using joint probability distribution in BN, intended query is solved using Equation 3.43.

$$P(U \mid X,Y) \propto \sum_{W}\sum_{V} P(X \mid W)P(Y \mid W)P(W \mid U,V)P(U)P(V) \tag{3.43}$$

Below we present step wise procedure of eliminating V and W from Equation 3.43

Step 1 : Eliminate V

τ_1(U, W) = $\sum_V$P(E)P(W | U, V)

Where τ_1(U, W) is a vector of values for each combination of U and W defined below:

τ_1(U, W) = P(V)P(W |U, V) + P($\bar{V}$)P(W |U, $\bar{V}$) = 0.940
τ_1(U, $\bar{W}$)= P(V)P($\bar{W}$ |U, V) + P($\bar{V}$)P(W' |U, $\bar{V}$) = 0.059
τ_1($\bar{U}$, W) = P(V)P(W |$\bar{U}$, V) + P($\bar{V}$)P(W |$\bar{U}$,$\bar{V}$) = 0.015
τ_1($\bar{U}$, $\bar{W}$) = P(V)P($\bar{W}$ |$\bar{U}$, V) + P($\bar{V}$)P($\bar{W}$ |$\bar{U}$, $\bar{V}$) = 0.998

The left hand side of Equation 3.43 can be rewritten as P(U)τ_1(U, W)P(X | W)P(Y | W)

Step 2 : Eliminate W

$\tau_2(\text{U, X, Y}) = \sum_U \tau_1(\text{U, W})\text{P(X} \mid \text{W)P(Y} \mid \text{W)}$

Since X and Y are observed as true and so τ_2(U, X, Y) is only a vector over the values of U and we use notation τ_2'(U) instead of τ_2(U, X, Y).

$\tau_2'(\text{U}) = \tau_1(\text{U, W})\text{P(Y} \mid \text{W)P(X} \mid \text{W)} + \tau_1(\text{U},\bar{W})\text{P(Y} \mid \bar{W})\text{P(X} \mid \bar{W}) = 0.592$
$\tau_2'(\bar{U}) = \tau_1(\bar{U}, \text{W})\text{P(Y} \mid \text{W)P(X} \mid \text{W)} + \tau_1(\bar{U}, \bar{W})\text{P(Y} \mid \bar{W})\text{P(X} \mid \bar{W}) = 0.001$

Using τ_2'(U) and P(U), P(U | X, Y) can be computed using Equation 3.44

$$P(U \mid X,Y) = \frac{P(U)\tau_2'(U)}{P(U)\tau_2'(U) + P(U')\tau_2'(U')} = 0.284 \tag{3.44}$$

The whole process took 19 multiplications, 7 additions and 1 division. However, same query using full joint distribution would take 128 multiplications alone.

3.8 Learning Bayesian Networks

This section is focussed on methodologies of learning parameters for variables encoded in the Bayesian model. In case where variables are discrete, parameters learned are conditional probability distributions whereas, for continuous variables density functions are learned. Broadly, there are two ways of learning parameters: (1) eliciting from domain experts and, (2) using learning methods to extract parameters directly from databases [48; 56]. Learning using domain experts could be problematic for several reasons such as their limited availability or the knowledge required is too large for expert's to find out time from their schedule. Learning methods using databases could provide a promising solution in this direction. In the information age, we get access to huge amount of data which could serve as a base for learning parameters for the model. For example, on a medical diagnose model, we may access to large amount of patient records listing attributes such as the patient's age, disease, symptoms, test results and more. Assuming we are given Bayesian graphical structure on this problem and set of records then, we may learn strength of relationship that exists among attributes using the database. For example, if variables disease and test results are directly connected

in the model then, using database we could define distribution such as probability of positive test result given disease is present.

The goal of parameter estimation is to find parameters values of a model that best fits the data. There are mainly two approaches to dealing with parameter-estimation in Bayesian network: (1) *maximum likelihood estimation (MLE for short)* and, (2) *Bayesian approach*. We discuss below general principles MLE approach in a discrete Bayesian framework.

3.8.1 Maximum Likelihood Estimation

We explain MLE using a simple example consisting of one random variable X, where X is the result of tossing a thumbtack. Let Ω_X = (Head (h for short), Tail (t for short)) represent set of possibilities over this problem. Further, suppose data set on tosses represented by D = (h, t, h, t, t, h, t) is given and, the task is to estimate P(X = h) denoted by θ. Considering tosses are independent, we may conclude that probability of the sequence is given by

P(h, t, h, t, t, h, t) = θ(1-θ)θ(1-θ)(1-θ)θ(1-θ) = θ^3(1-θ)^4

The probability of the sequence above depends on the particular value of θ. The *likelihood function* examines how the probability of the data changes as a function of θ. More precisely, the *likelihood function* is the probability of having observed the sequence given the parameter. We define *likelihood function* over given data set as below:

L(θ : (h, t, h, t, t, h, t)) = P((h, t, h, t, t, h, t) : θ) = θ^3(1-θ)^4

Let N_0 and N_1 denotes number of times H and T appears in the data set D respectively. Then, *likelihood function* defined above can be rewritten as Equation 3.45

$$L(\theta : D) = \theta^{N_0}(1-\theta)^{N_1} \tag{3.45}$$

N_0 and N_1 are called *sufficient statistics* [48] for the parameter θ as the likelihood depends only of the data through these values. Likelihood function defined above can be used a measure of quality for different parameters values to select the parameter value that maximizes the likelihood; this values is called *maximum likelihood estimator*

denoted by $\hat{\theta}$. For the data set D defined above, $\hat{\theta}$ = 0.42 = 3/7. Equation 3.45 can be rewritten in terms of logarithm called as *log-likelihood* for the ease of MLE computation, refer Equation 3.46. Differentiating the log-likelihood, setting the derivative to 0, and solving for θ we get the maximum likelihood parameter which is denoted in Equation 3.48

$$\ell(\theta : D) = N_0 \log \theta + N_1 \log(1 - \theta) \tag{3.46}$$

$$\hat{\theta} = \frac{N_0}{N_0 + N_1} \tag{3.47}$$

Formally, given data set D, MLE is a process of choosing parameter $\hat{\theta}$ which satisfies Equation 3.48.

$$L(\hat{\theta} : D) = \max_{\theta \in \Omega} L(\theta : D) \tag{3.48}$$

3.8.2 MLE for Discrete Bayesian Networks

In MLE it is assumed that Bayesian structure and a data set consisting of fully observed instances of the network variables are given in order to learn Bayesian parameters using MLE. Bayesian structure helps reducing task of parameter estimation into a set of smaller and unrelated problems, each of which can be solved using MLE principle described above. We present below using an example how Bayesian structure plays a key role in learning parameters for the model.

Consider a Bayesian network over three binary variables, X, Y and Z defined over relation structure, X $\rightarrow$ Y $\rightarrow$ Z. Let, Val(X) = (x_0, x_1), Val(Y) = (y_0, y_1) and Val(Z) = (z_0, z_1). Our goal in this example is to maximize the log-likelihood function defined over parameter θ, which defines the set of parameters for all CPDs in the network. That is, parameters: θ_{x_0}, θ_{x_1}, $\theta_{Y_0|X_0}$, $\theta_{Y_1|X_0}$, $\theta_{Y_0|X_1}$, $\theta_{Y_1|X_1}$, $\theta_{Z_0|Y_0}$, $\theta_{Z_0|Y_1}$, $\theta_{Z_0|Y_1}$ and $\theta_{Z_1|Y_1}$. For simplicity, we use the notation, $\theta_{Y|x^0}$ and $\theta_{Z|y^0}$ to refer to the sets $\{\theta_{y_0|x_0}, \theta_{y_1|x_0}\}$ and $\{\theta_{z_0|y_0}, \theta_{z_1|y_0}\}$ respectively. Also, we use short notations for $\theta_{Y|X}$ and $\theta_{Z|Y}$ to refer to $\{\theta_{Y|x_0} \cup \theta_{Y|x_1}\}$ and $\{\theta_{Z|y_0} \cup \theta_{Z|y_1}\}$ respectively. The likelihood function over data set instance, {x[m], y[m], z[m]} on this example would be:

L(θ : D) = $\prod_{m=1}^{M}$ P(x[m], y[m], z[m] : θ)

Using Bayesian structure into account, P(x[m], y[m], z[m]) can be rewritten as a product form, P(z[m]|y[m])P(y[m]|x[m])P(x[m]). Using product form and exchanging the order of multiplication, likelihood can be decomposed as defined in Equation 3.49.

$$L(\theta : D) = \left(\prod_m P(x[m] : \theta)\right)\left(\prod_m P(y[m]|x[m]) : \theta\right)\left(\prod_m P(z[m]|y[m]) : \theta\right) \quad (3.49)$$

As indicated in Equation 3.49, likelihood decomposes into three separate terms, one for each variable. Each term is then maximized using Equation 3.48 independently and, then all components are combined in order to get an MLE solution. For example, the second component in Equation 3.49 can be further decomposed as shown by Equation 3.50 into two factors in order to calculate its local maxima. For every initiation of X, $\theta_{y_1|x_0}$ is further computed using Equation 3.51.

$$\prod_m P(y[m] \mid x[m] : \theta_{Y|X}) = \prod_{m:x[m]=x_0} P(y[m] \mid x[m] : \theta_{Y|x_0}) \prod_{m:x[m]=x_1} P(y[m] \mid x[m] : \theta_{Y|x_1}) \quad (3.50)$$

$$\theta_{y_1|x_0} = \frac{M[x_0, y_1]}{(M[x_0, y_1] + M[x_0, y_0])} \quad (3.51)$$

Concluding the discussion above, for each variable X_i in Bayesian network, likelihood is defined using Equation 3.52. This shows that the likelihood decomposes as a product of independent terms, one for each CPD in the network . This property is called the global decomposition of the likelihood function [48].

$$L_i(\theta_{X_i}|Pa(X_i) : D) = \prod_m P(X_i[m] \mid Pa(X_i)[m] : \theta_{X_i}|Pa(X_i) \quad (3.52)$$

3.8.3 Learning Bayesian structure

There are two major approaches of existing structure learning methods: *constraint based approaches* and *score-based approaches*.

Constraint-based approaches first attempt to identify a set of conditional independence properties, and then attempt to identify the network structure that best satisfies

these constraints. These approaches are quite intuitive in the sense that they decouple the problem of finding structure from the notion of independence, following very close to the definition of Bayesian network. The drawback with the constraints based approaches is that it is difficult to reliably identify the conditional independence properties and to optimize the network structure [53]. Plus, these methods can be sensitive to failures in individual independence tests. If any of the tests return a wrong answer then it may mislead the network construction procedure [48]. The two most popular constraint-based algorithm are the SGS algorithm and PC algorithm proposed by Spirtes et al. [67]. SGS algorithm determines the existence of an edge between every two node variables by conducting a number of independence tests between them conditioned on all the possible subsets of other node variables. The PC algorithm is a more efficient constraint-based algorithm. It conducts independence tests between all the variable pairs conditioned on the subsets of other node variables that are sorted by their sizes, from small to large. The subsets whose sizes are larger than a given threshold are not considered. More detail on SGS and PC algorithm can be found here [53] [67].

Score-based approaches first define a score function indicating how well the network fits the data, then search through the space of all possible structures to find the one that has the optimal value for the score function. Problem with this approach is that it is intractable to evaluate the score for all structures, so usually heuristics, like greedy search, are used to find the sub-optimal structures. Score-based approaches are typically based on well established statistical principles such as minimum description length (MDL) [49] or the Bayesian score [61]. The MDL criterion requires choosing a network that minimizes the total description length of the network structure and the encoded data, which implies that the learning procedure balances the complexity of the induced network with the degree of accuracy with which the network represents the data. The K2 algorithm [22] is an example of a score-based approach.

3.9 Causality and Bayesian Networks

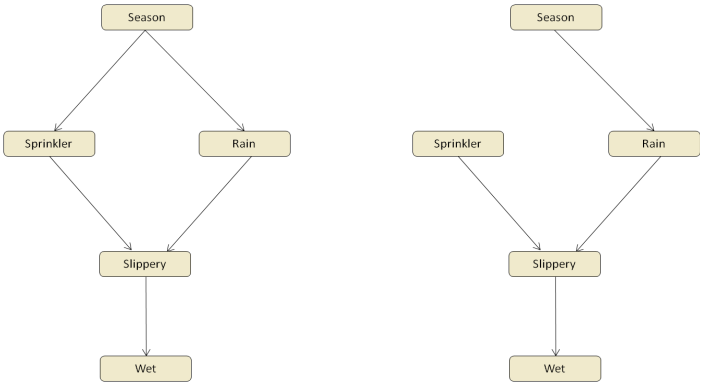
Causality in simple terms defines relationship between causes and effects. So far in this chapter, we have described Bayesian network in *cause-effect* framework which is indicated by a directed arrow connecting two variables. However, being directed, does not simply imply that there exists a causal relation. For example, Cold $\rightarrow$ Headache

is a causal network whereas Headache → Cold is not, even though both networks are equally capable of representing any joint distribution on the two variables. The graphical structure of the Bayesian network can still induce concepts of joint probability, conditional independence assumptions and probabilistic queries regardless of the fact whether direction of arrows are meaningful or not. However, it is a common wisdom that a good Bayesian structure corresponds to causality [48].

A causal model has the same form as probabilistic Bayesian network. It consists of a directed acyclic graph over the random variables in the domain. The model asserts that each variable X is governed by a causal mechanism that determines its value based on the values of its parents. In other words, it is assumed in a causal model that causality flows in the direction of the edges. However, unlike basic belief propagation for solving probabilistic queries in Bayesian networks, causal analysis goes one step further by inducing reasoning about situations where we *intervene* in the world. The aim of intervening is to infer not only beliefs or probabilities under static conditions, but also the dynamics of beliefs under changing conditions, for example, changes induced by treatments or external interventions like, planning everyday activity. The intervention queries are of forms for example: P(Y | do(z)) or P(Y | do(z), X = x). Where do(Z = z) corresponds to setting where an agent directly manipulated the world to set the variable Z to take the value z with probability 1. We present below example from [58] to explain concept of intervention queries.

Example: A simple BN in Figure 3.15a describes relationship between five variables. Each variable is binary in nature except the variable season which takes four states. Suppose in this BN, we represent the action "turning the sprinkler on". In order to perform this action, we delete all incoming arcs in variable Sprinkler and, set its value to "on". The result of this action is the BN shown in Figure 3.15b. The deletion of the arc Season → Sprinkler gives the understanding that whatever relationship existed between Season and Sprinkler prior to the action, that relationship no longer in effect while the action is performed. There is a difference between setting observation, Sprinkler = on and do(Sprinkler = on). The effect of first observation is obtained by Bayesian conditioning, P(Season, Rain, Wet, Slippery | Sprinkler = on) while, that of later by conditioning a mutilated graph with the link Season → Sprinkler removed (refer BN in Figure 3.15b).

A causal model follows following two key assumptions [48]:



(a) Bayesian network with five variables

(b) Representation of action "turning the Sprinkler on" on Bayesian network

Figure 3.15: Causality and Bayesian network

Causal Markov assumption: This assumption asserts that each variable is conditionally independent of its non-effects given its direct causes. Thus, each variable is conditionally independent of its nondescendants given its parents.

Faithfulness assumption: It states that conditional independence assumptions are those that arise from the d-separation in the corresponding Bayesian structure.

The intuition behind Causal Markov assumption is that if we ignore variable's effects then, all the relevant probabilistic information about a variable is contained in its direct causes. On the other hand, Faithfulness assumption asserts that all conditional independence conditions incurred in the causal model are consequences of Causal Markov (or d-separation) condition. In other words, a causal model assumes that whatever independencies arise are not due to coincidence but rather are induced because of structure.

In Bayesian network, Causal Markov assumption is same as one defined in definition 3.1 except that arcs are given a causal interpretation. The ability of Bayesian network to encode directional relations which can represent *cause-effect* relationships as compared to other graphical model for example, Markov models that cannot stands for an important reason of considering Bayesian networks as a causal models. In addition to this, assumption of Faithfulness is reasonable and widely embraced in practice for graphical model such as Bayesian networks [48; 58].

3.10 Study on Bayesian Network Softwares and Packages

The rapid development of Bayesian network research over years has been accompanied by a proliferation of Bayesian network software tools. These tools are built to apply capability of Bayesian knowledge representation scheme and reasoning over wide range of application domains. This section presents a comprehensive study of major software and packages dealing with Bayesian networks. We present discussion on following Bayesian softwares and packages: Netica, Hugin, Bayes Server, GeNIe & Smile, SamIam, B-Course, DEAL (package in R), Bnlearn (package in R) and Bayes Net toolbox (package in Matlab). We endeavor to point out important features of these softwares and packages such as: GUI and API, Bayesian learning, inference, whether software is free or commercial and types of nodes supported (i.e., discrete and/or continuous).

Netica

Netica [24] is the most widely used commercial Bayesian development software from Norsys software corp. This software provides an intuitive and smooth user interface for drawing the networks. The GUI is available for Mac and Windows. The relationships between variables may be entered as individual probabilities, in the form of equations, or learned from data files. Netica supports text files, CSV formats or an ODBC connections in order to learn parameters for a defined structure using expectation-maximization algorithm. The software version 5.0 and later introduced Bayesian structure learning using Tree-augmented naive (TAN) Bayes approach. Netica can use the networks to perform various kinds of inference (causal and diagnostic) using the fastest and most modern algorithms. The change in probabilities corresponding to nodes can be displayed in a number of different ways, including bar graphs and meters. In addition to basic inference tasks, Netica also supports probabilistic queries such as most probable explanation and sensitivity analysis. This application also allows the import of Bayesian network in the several formats for example, “DSC”, “XML”, “NET” and more. The Netica API is available in several languages for example, C, Java, Matlab, C# to run on Mac, Linux and Windows. The Bayesian network repository [23] containing examples

of popular Bayesian networks is also maintained by Netica.

Hugin

Like Netica, Hugin [29] is also one of the popular commercial product for Bayesian development and analysis. The original Hugin shell was initially developed by a group at the Aalborg University in 1989, and now it is owned by Hugin Expert, Ltd. Hugin provides an interactive tool for creating and manipulating Bayesian models. The Hugin API is called "Hugen Decision Engine". It is available for the languages C++, Java and as an ActiveX-server and runs on the operating systems: Sun Solaris, Linux and Windows. HUGIN Decision Engine implements state-of-the-art algorithms for Bayesian networks and influence diagrams such as object-oriented modeling, learning from data with both continuous and discrete variables, value of information analysis, sensitivity analysis and data conflict analysis. The conditional probability tables (CPTs) can be specified with expressions as well as manual entry. The CPTs do not have to sum to one; entries that do not sum to one are normalized. The parameter learning in Hugin is with the use of expectation-maximization algorithm while learning structure is ensured with the use of two constraint-based algorithms PC and NPC.

Bayes Server

Bayes Server [51] owned by Bayes Server Ltd. made first public release in year 2008. The software specializes intelligent systems, such as those found in machine learning and artificial intelligence. Bayes Server can be used to build Bayesian networks and Dynamic Bayesian networks to perform tasks such as Classification, Regression, Time Series prediction, Segmentation/Clustering, Density estimation, anomaly detection, Decision Support, reasoning, multivariate data analysis and much more. It includes a user interface and API for building and visualizing models, learning models from data, sampling data, charting, and building complex probability queries, including time series predictions. Bayes Server supports continuous variables using Conditional Gaussian distributions. Support for continuous variables is also included for Dynamic Bayesian networks (time series). Bayes Server supports parameter learning however, Bayesian structure learning is in progress. The Bayes Server libraries are restricted for languages

that can be interfaced with .NET platform. The software support several charts for visualization. The lift chart can be used to measure the performance of a Bayesian network when it is used for classification. Discrete and continuous histograms can be generated based on given data.

SamIam

SamIam [3] was established in year 2004 by the Automated Reasoning group at university of California Los Angeles. SamIam stands for Sensitivity, Analysis, Modeling, Inference And More. It is comprehensive tool for modeling and reasoning with Bayesian networks developed in Java language. It supports a graphical user interface for drawing and inferencing models. SamIam is free software which includes several algorithms for Bayesian inference, maximum probable explanation and sensitivity analysis. The parameter learning is supported by the software while, structure learning algorithms are not included in the software yet. The software can be used for loading and saving Bayesian network models in a variety of file formats. While modeling Bayesian networks, only discrete nodes can be used. The continuous data types are not supported by SamIam.

B-Course

B-Course [6] is a free web based data analysis tool for Bayesian modeling, in particular dependence and classification modeling. The software service is hosted by Complex Systems Computation Group CoSCo, Helsinki Institute for Information Technology in year 2002. The software offers two types of modeling: dependency and, classification. In the dependency modeling, B-Course finds the model of the probabilistic dependencies among variables given in the data set. Besides revealing the structure of the domain from data, dependency models can be used to infer probabilities of any set of variables given any (other) set of variables. On the other hand, classification modeling demonstrate how to build a simple classification model out of a data set, and how to use it for predicting the class membership of unclassified data. Once Bayesian model is built by the software, the model can be saved and downloaded. In addition to standard Bayesian structure modeling, B-Course also offer two graphical representations describing the possible causal relationships that may have caused the dependencies in

the model. These causal graphs are based on the calculus introduced by Pearl [58]. To best of knowledge, this feature is unique to B-Course.

Bayes Net Toolbox (BNT)

Bayes Net Toolbox [42] is an open source library for use with only Matlab, a widely used and powerful mathematical software package. BNT was developed by Kevin Murphy in 2001. The package includes several algorithms for inference and learning Bayesian networks. Package lack support of GUI however, few visualizations are possible due to Matlab's features. BNT supports both discrete and continuous variables. The conditional probability tables are represented in a tabular format in case of discrete nodes while, gaussian distribution is used to represent continuous variables. BNT offers several inference algorithms for Bayesian networks including discrete, Gaussian and mixed data types (conditional Gaussian). For learning parameters, BNT uses two types of learning. First learning setting is based on maximum likelihood for complete data, and second uses expectation maximization algorithm for incomplete data. This package also supports Bayesian structure learning algorithms namely K2, MCMC and PC. However, structure learning feature of this package is suitable only for low dimensional data set. BNT also support implementation of dynamic models such Hybrid Markov Models, Dynamic Bayesian Models and Kalman Filters.

GeNIe & SMILE

GeNIe and SMILE [33] are developed by the Decision Systems Laboratory (DSL), School of Information Sciences, University of Pittsburgh in 1998. The software consist of two modules: GeNIe and, SMILE. GeNIe (graphical network interface) is an environment for the decision and the construction of Bayesian networks characterized by its inference engine SMILE (structured modeling reasoning and learning engine). The inference engine SMILE, consist of a library of C++ classes compiled for Windows, Solaris and Linux. This software does not support continuous variables hence, any continuous variable is discretized before modeling the network. The software support parameter and structure learning algorithms. It supports several backup formats "xDSL", "DSL", "NET", "DNE", "DXP" and "DSC". Bayesian network repository [34] is also managed on the website of this software.

Deal package in R

Deal [32], software package was developed by Bottcher and Dethlefsen for learning Bayesian structure in R. It includes several methods for analysing data using Bayesian networks with variables of discrete and/or continuous types but restricted to conditionally Gaussian networks. This package do not support Bayesian inference. However, it provides interface to Hugin [29]. The GUI of Hugin can then be further used for inference task. Deal supports visualization of learnt network. For this package, user can define set of direct relationships which are not allowed.

Bnlearn package in R

Bnlearn [62] is an R package for learning the graphical structure of Bayesian networks, estimate their parameters and perform some useful inference. The package was developed by Marco Scutari in 2009. The package supports several constraints and score based Bayesian structure learning algorithms. The continuous data type variables are not allowed by the package. This package does not support visualization of the learnt Bayesian model. However, it generates a detail report on learnt model which includes set of directed nodes, BIC score of the network and may more. The package includes collection of popular Bayesian networks [63].

Comparison of Bayesian network software and packages

In Table 3.3 we summarize technical comparison of softwares and packages manipulating Bayesian networks discussed above. We discuss features: GUI (Yes if GUI is supported; No otherwise), API (Yes if GUI is supported; No otherwise), whether software supports continuous variables (Both if both discrete and Continuous variables are supported; Discrete otherwise), whether inference is supported (Yes if inference is supported; No otherwise), whether software offers parameter learning (Yes if supported; No otherwise), whether software offers structure learning (Yes if supported; No otherwise) and whether the software is licensed (Free if software is free; Licensed otherwise).

Name	**GUI**	**API**	**Variables**	**Inference**	**Parameter learning**	**Structure learning**	**License**
Netica	Yes	Yes	Discrete	Yes	Yes	Yes	Licenced
Hugin	Yes	Yes	Both	Yes	Yes	Yes	Licenced
Bayes Server	Yes	No	Both	Yes	Yes	Yes	Licenced
SamIam	Yes	No	Discrete	Yes	Yes	No	Free
B-Course	Yes	No	Discrete	No	Yes	Yes	Free
BNT	No	Yes	Both	Yes	Yes	Yes	Free
GeNIe & SMILE	Yes	No	Discrete	Yes	Yes	Yes	Licenced
Deal	Yes	Yes	Both	No	Yes	Yes	Free
Bnlearn	No	Yes	Discrete	Yes	Yes	Yes	Free

Table 3.3: Comparison of Bayesian network softwares and packages

Chapter 4

Mining Anomalies Using Bayesian Networks

This chapter is based on following publications:

1. Integration of Domain Knowledge for Outlier Detection
   in High Dimensional Space
   Sakshi Babbar
   In Database Systems for Advanced Applications (DASFAA) Workshop,
   Brisbane, Australia, 2009, pp. 363-368

2. On Bayesian Network and Outlier Detection
   Sakshi Babbar and Sanjay Chawla
   In proceedings of the 16th International Conference on Management of Data ,
   Nagpur, India, 2010, pp. 125-137

3. A Causal Approach For Mining Interesting Anomalies
   Sakshi Babbar, Didi Surian and Sanjay Chawla
   In proceedings of the 26th Canadian Conference on Artificial Intelligence,
   Regina, Canada, 2013, pp. 226-232

4.1 Introduction

An outlier is a data instance in a database which is significantly different from the norm. The objective in outlier detection, is not only to identify outliers in large and high dimensional databases but also to *correlate* them with actual anomalous events. For example, if the outlier detection techniques are being used for finding anomalies in network traffic, then outliers in network data should correspond to physical anomalies - like denial of service attack or ping flood. Thus if O is a set of discovered outliers from data and A is the set (unknown) anomalies, then an ideal good outlier detection method will have high precision and recall, i.e., both $P(A|O)$ and $P(O|A)$ are high. The challenge in outlier detection is that we rarely, if ever, have access to the anomalous set A. Thus like clustering, outlier detection is an unsupervised learning method.

Current data mining methods identify sparse regions in point cloud data to search for outliers. For example, in distance-based methods, a data point is an outlier if it is effectively far away from its neighbors. Variations on distance-based approaches, like those based on density, incorporate the local density of the region while reporting outliers, though the principle remains the same. However, as we will demonstrate, such approaches ignore valuable information that is available in the data.

Suppose we conceptually place a fine resolution grid on the point cloud space. For example, in an N-dimensional data set we can identify the grid cells with the a lattice Z^n. Now, distance-based outliers are essentially data points which live in sparse cells. In fact we can associate a probability with each cell, which is the percentage of data points which lie in that cell. In the language of pattern mining, cells with low (but non-zero) *support* contain the outliers. A major objective of this paper is to show that when we want to search for outliers and then use them to identify anomalous events, then the focus on *confidence* yields more meaningful results.

In this chapter, we propose a novel approach which combines the use of Bayesian network (BN) and probabilistic association rules to discover and explain anomalies in data. The Bayesian network allows us to organize information in order to capture both correlation and causality in the feature space, while the probabilistic association rules have a structure similar to association mining rules. In particular, we focus on two types of rules: (i) *low support & high confidence* and, (ii) *high support & low confidence*. The measures *support* and *confidence* can also be addressed as prior and conditional probability respectively in BN. However, unlike traditional association rule mining we

are not in pursuit of mining frequent patterns using these measures but instead, we are interested in mining infrequent patterns whose occurrence suggests the presence of uncommon and exceptional situations. We refer the discovered anomalous patterns as *Domain Specific Anomalous patterns* (or **DSAPs**). In order to test if a particular test case is an anomaly for a given domain, we check if it carries "any" pattern from the discovered set of DSAPs. We address our method as a *Causal Outlier Mining* (**COM**) approach.

In addition to proposing a new approach for anomaly detection, we also present critical analysis on: (1) the search methodology of distance based techniques (2) Bayesian approach and, (3) why a data point discovered as an outlier by a distance based technique is not necessarily an outlier from the Bayesian perspective.

4.1.1 Problem Statement

The problems that we address in this chapter are as follows:

1. Given set of data points, report and explain those anomalous data points which are both interesting and useful for domain.

2. Why traditional distance based techniques may not be an accurate and effective technique to discover true anomalies?

4.1.2 Contributions

We describe our contributions are as follows:

1. We propose a novel approach that combines the use of Bayesian network and probabilistic association rules to discover anomalies in data. We focus on the *causality* effect that describes why an observation is anomalous.

2. Our proposed approach is designed specifically to give contextual information of an anomaly, which could also be used to enrich our knowledge about anomalies.

3. We also present critical analysis of distance based techniques which highlights why distance based criteria may not be an accurate and effective technique to discover true outliers.

4. We perform extensive experiments and show that our proposed approach gives results in high precision and recall.

4.1.3 Notations and Basic Concepts

In this chapter all notations corresponding to Bayesian networks are followed from Chapter 3. However, notations specific to this chapter are summarized in Table 4.1.

Notation	Description
d	A data point
DSAPs	Domain specific anomalous patterns
$\|$DSAPs$\|$	Total number of DSAPs extracted
τ	Number of DSAPs selected
Z	Topic assignment for a word
W	Word
K^*	Number of topic
V	Total number of vocabularies
β	Word distributions for topics
α	Hyperparameter
M	Total number of documents
N	Total number of words in document
θ	Topic mixture of a document

Table 4.1: Notations and basic concepts

The remainder of the chapter is organised as follows: in Section 4.2, we present our detailed methodology. Our experiments and analysis on results are explained in Section 4.3. Finally, in Section 4.4 we conclude the chapter.

4.2 COM Methodology

In this section, we explain two probabilistic rules which we address as $\mathbf{R}_1$ and $\mathbf{R}_2$ to mine interesting low probable patterns from a given domain whose knowledge is captured by a Bayesian network. These rules are applied in each causal interaction of the form P(X | Pa(X)) encoded in the model. We call these causal interactions as *causal subspaces*. Studying each causal subspaces gives advantage of mining anomalies in a subspace level through which reasons of anomalous nature can also be explained. Consider Bayesian network in Figure 4.1. There exist two causal interactions, i.e., ($X_2 \mid X_1$,

X_3) and ($X_3 \mid X_1$). In general, we write these causal subspaces as (X_1, $X_3 \rightarrow X_2$) and ($X_1 \rightarrow X_3$).

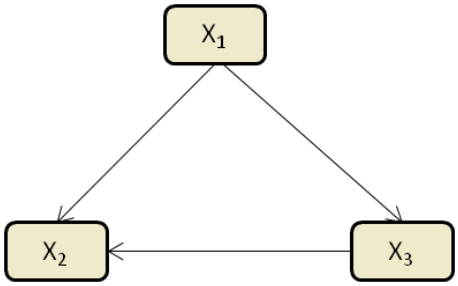


Figure 4.1: Bayesian network encoding two causal subspaces: (X_1, $X_2 \rightarrow X_3$) and ($X_1 \rightarrow X_3$)

Before proceeding, we make following clarification on theory of causal subspaces and rules.

- Rules are applied on each causal subspace encoded in the Bayesian network in order to reveal low probable patterns residing in subspaces.
- A parent node in one causal subspace could appear as a child node in another causal subspace and vice-versa. For example BN in Figure 4.1, X_3 is a parent node in causal subspace (X_1, X_2, X_3) while, child node in causal subspace (X_1, X_3).
- In any causal subspace there could exist more than one parent of a child node but, child node more than one is not possible.

With this clarification, we now define $\mathbf{R}_1$ and $\mathbf{R}_2$ as follows:

1. $\mathbf{R}_1$: *In every causal subspace, select that state in child node which has a high confidence conditioned on all its parents in low support.*
2. $\mathbf{R}_2$: *In every causal subspace, select those state(s) in child node which have a low confidence conditioned on all its parents in high support.*

Both of the these rules work on principle of two measures namely *support* and *confidence*. The definitions of support and confidence of a variable in BN is defined using Equation 4.1 and 4.2. Support of a variable X is like a prior probability in some

state of x_i. In contrast, confidence is a conditional probability of a variable X in some state x_i given set of observations on its parent nodes.

$$support(X = x_i) = P(X = x_i) \tag{4.1}$$

$$confidence(X = x_i) = P(X = x_i \mid Pa(X)) \tag{4.2}$$

For an example illustration on definition of *support* and *confidence* defined by Equations 4.1 and 4.2 respectively, consider BN in Figure 4.2. Based on unconditional and conditional probabilities associated with variables, example of *support* in variable Sun are *support(Sun = T) = 65%* and *support(Sun = F) = 35%*. Whereas, few examples of *confidence* on conditional variable Rain are: *confidence(Rain = T | Sun = T) = 90%* and *confidence(Rain = F | Sun = F) = 80%*.

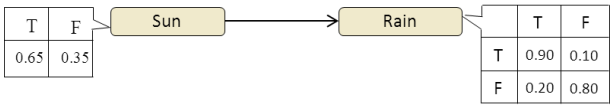


Figure 4.2: A three-node Bayesian network. One example of measure *support* in parent node Sun is: *support(Sun = T) = 65%* whereas, example of measure *confidence* in child node Rain is: *confidence(Rain = T | Sun = T) = 90%*

In our work, we use concept of support for all parent nodes in each causal subspace structured in the Bayesian network whereas, confidence is computed for each child node encoded in the causal subspace. This implies, Equation 4.1 and Equation 4.2 can be rewritten as Equation 4.3 and Equation 4.4 respectively for each causal subspace encoded in the Bayesian model.

$$support(X = x_i)_{X \in CS_j} = P(X = x_i)_{X \in CS_j} \tag{4.3}$$

$$confidence(X = x_i)_{X \in CS_j} = P(X = x_i \mid Pa(X))_{X,\, Pa(X) \in CS_j} \tag{4.4}$$

Intuitively, rules $\mathbf{R}_1$ and $\mathbf{R}_2$ mine those suspicious patterns which do not provide enough evidence to accept them as an usual theory of the domain but, actually are indicator of an alternative theory not favored by the domain. The $\mathbf{R}_1$ focuses on the extraction of the *"low support & high confidence"* patterns, which refers to the patterns

whose "cause" appears with low probability, but interestingly the impact on the "effect" is strong. On the other hand, the rule $\mathbf{R}_2$ aims for the *"high support & low confidence"* patterns, which means that $\mathbf{R}_2$ mines those patterns whose "cause" appears with high probability, but has low impact on the respective "effect". We exclude *"low support & low confidence"* patterns because the causal relationship that showing low conditional probability conditioned on low prior is more likes a *noise* rather than an anomaly.

We refer the low support, high support, low confidence, and high confidence as *minsupp*, *maxsupp*, *minconf*, and *maxconf* respectively. The first two are Bayesian specific, while the last two are parameters defined by a user. Equation 4.5 and Equation 4.6 defines the mathematical definitions for *minsupp* and *maxsupp* respectively.

$$minsupp(X = x_i)_{X \in CS_j} = x_i \text{ s.t. } \min_i (P(X = x_i))_{X \in CS_j} \text{ holds for } x_i \tag{4.5}$$

$$maxsupp(X = x_i)_{X \in CS_j} = x_i \text{ s.t. } \max_i (P(X = x_i))_{X \in CS_j} \text{ holds for } x_i \tag{4.6}$$

Application of these rules in each causal subspace of BN results in mining DSAPs which has an implication expression of the form:

$$X[x_i] \rightarrow C[c_j] \tag{4.7}$$

where the left hand side of the arrow represents parent nodes and the right hand side of the arrow is their respective child node. Information enclosed in the angular braces represents states satisfying rules taken by parent and child nodes respectively. Equation 4.8 and Equation 4.9 present the formal definitions of these rules.

$$\mathbf{R_1}: \forall X \in Pa(C) \in CS_j \; s.t. \; (P(X = x_i) = minsupp) \wedge (P(C = c_k \mid X) > maxconf) \tag{4.8}$$

$$\mathbf{R_2}: \forall X \in Pa(C) \in CS_j \; s.t. \; (P(X = x_i) = maxsupp) \wedge (P(C = c_k \mid X) < minconf) \tag{4.9}$$

We present a small hypothetical Bayesian network in Figure 4.3 to give more understanding on how anomalous patterns are extracted in practice using rules $\mathbf{R}_1$ and $\mathbf{R}_2$.

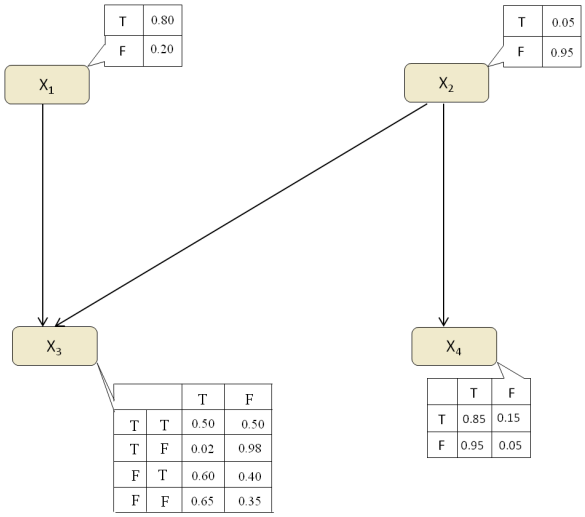


Figure 4.3: Hypothetical Bayesian network showing unconditional and conditional probabilities associated with each node

Figure 4.3 shows a BN with four nodes namely, X_1, X_2, X_3 and X_4. Let each of these nodes takes two distinct states say, True (T) and False (F). Associated with each node is unconditional probability table for parent nodes (X_1, X_2) and conditional probability table for child nodes (X_3, X_4). Following Equation 4.5 and 4.6, *minsupp* and *maxsupp* for parent nodes X_1 and X_2 are set to:

$$minsupp(X_1) = F,\ minsupp(X_2) = T,\ maxsupp(X_1) = T \text{ and } maxsupp(X_2) = F$$

Suppose the parameters *minconf* and *maxconf* are set to 10% and 80% respectively. We apply rules on two causal subspaces, (X_2, X_4) and (X_1, X_2, X_3) encoded in this BN. In the first causal subspace application of $\mathbf{R}_1$ and $\mathbf{R}_2$ results in mining two DSAPs namely, $X_2[T] \rightarrow X_4[T]$ (example of $\mathbf{R}_1$) and $X_2[F] \rightarrow X_4[F]$ (example of $\mathbf{R}_2$); while in the second causal subspace only one DSAP, $X_1[T], X_2[F] \rightarrow X_3[T]$ is present (example of $\mathbf{R}_2$). The rule $\mathbf{R}_1$ is not applicable in this specific causal subspace since it does not qualify condition stated by Equation 4.8. Using these rules we discovered in total three DSAPs for this BN.

Assume that the test data is given for the domain on which this BN is formed and the objective is to discover the anomalous test cases. For this task we simply check if a test case carries any of the discovered DSAP. A simple and straight forward approach we followed is by forming a SQL SELECT query of extracted DSAPs. Each DSAP can

be thought as an individual component in WHERE clause of SQL SELECT query separated by an OR operator and within DSAP each item separated by an AND operator. For example, SQL SELECT query for DSAPs discovered on example above would be:

*SELECT * from test set where (X_1='T' AND X_2='F' AND X_3='T') OR (X_2='T' AND X_4='T') OR (X_2='F' AND X_4='F')*

We now take income-expenditure example discussed in Chapter 1 to show how data points in clusters C_5 and C_6 can be mined using rules. Suppose we form a Bayesian network on this problem, refer Figure 4.4. Let each variable take up two states namely, high (H) and low (L). Associated with variables are unconditional and conditional probabilities representing a rough scenario of data points presented in Figure 1.1 of Chapter 1. If *minconf* = 10% and *maxconf* = 80%, then based on Equations 4.8 and 4.9, an interesting DSAP would be: Income[L]→ Expenditure[H] ($\mathbf{R}_2$). The scenario of low income and high expenditure is represented by clusters C_5 and C_6, and hence all data points forming these clusters will be identified as outliers using COM approach.

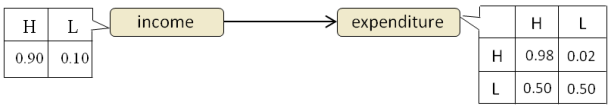


Figure 4.4: Bayesian network on income-expenditure example in a discrete framework

Our assumption that considers each DSAP as an indicator of anomalous event may lead to high false positive rate because of multiple hypothesis testing problem especially in the case when total number of DSAPs (*denoted by notation* $|DSAPs|$) from a BN is large. In order to control false positive rate, we propose to rank extracted DSAPs on how interesting they are from the Bayesian perspective only if condition: $|\text{DSAPs}| > 2 * |\mathbf{X}|$ is satisfied. We apply the concept of *sensitivity analysis* [40] in Bayesian networks, which is a measure of how sensitive is the conclusion to the findings for ranking discovered DSAPs. Sensitivity analysis in BN is performed by entering the known observations and studying sensitivity incurred in variable of interest. If the findings give negligible impact on a node under study, then the findings are considered sufficiently influential. On the other hand, if the impact on a node under study is significant, then those observations are considered least interesting for the investigated node. To score

every extracted DSAP on a sensitivity measure, observations in variables present on the left hand side of the arrow in the DSAP are entered in the model and sensitivity for the variable on the right side of arrow is computed. We then sort DSAPs in an ascending order and consider the top τ patterns with the lowest scores as the most interesting unlikely patterns.

4.2.1 Algorithm

We present the causal outlier mining in Bayesian network (**COMBN**) algorithm in Algorithm 1.

Algorithm 1 COMBN

Input: BN, parameters *minconf*, *maxconf*, $|\mathbf{X}|$, τ and a test set
Output: DSAPs, anomalies
1. Compute *minsupp* and *maxsupp* for every parent node in BN using Equations 4.5 and 4.6
2. For all causal subspace in BN, repeat:
   2.1. Apply $\mathbf{R}_1$ and $\mathbf{R}_2$ using Equations 4.8 and 4.9 to discover DSAP
   2.2. Compute sensitivity of discovered DSAP in BN
3. If ($|\text{DSAPs}| > 2 \times |\mathbf{X}|$) then,
   3.1 Sort DSAPs
   3.2 Output top (τ * $|DSAPs|$) low scored DSAPs
   else
   Output all DSAPs extracted
4. Output test cases with DSAPs within as anomalies

We explain algorithm COMBN with the help of Bayesian network presented in Figure 4.3. This BN can be considered as a model for the domain where objective is to identify outliers from given test set based on knowledge captured by the model. As an input we are given Bayesian network, parameters *minconf*, *maxconf*, τ and a test set. Let parameters *minsupp*, *maxsupp* and τ are set to 10%, 80% and 50% respectively. Algorithm starts with computing *minsupp* and *maxsupp* for all parent nodes in the model using Equations 4.5 and 4.6 as indicated by the step 1 in COMBN. Thereafter, rules $\mathbf{R}_1$ and $\mathbf{R}_2$ are applied over two causal subspaces present in this BN to discover DSAP. For every DSAP extracted using rules, its sensitivity score is computed. The total number of DSAPs extracted from this BN is three (it is discussed before). Since $|DSAPs|$ is less than $|\mathbf{X}|$ present in the BN so condition specified in step 3 of the algorithm is not satisfied and hence all three DSAPs extracted are given as output. Further, test cases with the presence of any of the three DSAP within are identified as outliers.

The computational complexity of the algorithm COMBN is governed by two key components in BN, i.e., (1) qualitative component, which specifies the number of nodes and directed links that present in the model and, (2) quantitative component, which indicates the total number of unconditional and conditional probability entries in the BN. Major computation involved are in Step 1 & 2 of the algorithm COMBN. In Step 1 of the algorithm, for every parent node, *minsupp* and *maxsupp* are maintained. However, to compute the state for which probability of occurrence is minimum and maximum for each parent node, we are not inferencing in Bayesian network which is known to be a NP-hard problem [48]. These parameters are like prior probabilities, i.e., $P(X = x_i)$, either provided by a domain expert or is learnt using EM algorithms from a given data set. Bayesian network development software like Netica [24] maintains this information for every node in the Bayesian network. Assuming this information is given, we only need to sort $P(X = x_i)$. Later *minsupp* and *maxsupp* are set using Equations 4.5 and 4.6 for every parent node in the BN.

In Step 2.1 of the algorithm, we use rules $\mathbf{R}_1$ and $\mathbf{R}_2$ in every causal subspace of a given Bayesian network to mine anomalous patterns. Intuitively, these rules are like finding conditional probability in some state of child node, given observations on parent nodes, i.e., $P(C = c_i \mid Pa(C))$. Interestingly, for queries like $P(C = c_i \mid Pa(C))$, again, we do not need any complex inference in Bayesian network. Rather information on such query is already pre-computed in BN in the form of conditional probability table associated with every child node. Query such as, $P(C = c_i \mid W)$ where, W belongs to set of descendent nodes of C in BN may require operations such as, marginalization over irrelevant variables for computing such probability of interest. Computing for such queries can go intractable if there are a large number of nodes in the Bayesian network. In comparison, query $P(C = c_i \mid Pa(C))$ is always tractable. However, a large conditional probability table could be a time consuming job in finding conditional probability of interest. In order to avoid such circumstances, we designed pruning strategy especially for rule $\mathbf{R}_1$. In rule $\mathbf{R}_1$, we are in pursuit of finding that entry in the conditional probability table where confidence is greater than or equal to *maxconf* threshold. For example, consider variable X with $|Val(X)| = 3$. On setting *maxconf* to 70%, we can find only one conditional probability in X greater than 70%. This condition holds true for all child nodes in BN. As soon as we find that entry, we break the scanning process in conditional probability table associated with child node since there would be only one entry greater than *maxconf* threshold. For rule $\mathbf{R}_2$, we need to scan probabilities

in all possible states of child node for given observations for entries less than *minconf* threshold since there could be more than one value satisfying *minconf* threshold. It can be imagined as scanning matrix of size (1 x |Val(C)|) where, |Val(C)| represents number of states of child node C.

In Step 2.2, the interestingness of a DSAP is computed using a sensitivity analysis in BN. Sensitivity analysis, again is a NP-hard problem in the worst case. However, in our work we use this measure for variables which are causally related, i.e., in every causal subspace rather than using this measure where known observations are sparsely located from the node on which sensitivity has to be analysed. Thus, sensitivity analysis is not NP-hard in our case.

4.3 Experiments, Results and Discussion

In this section we report on experiments that we carried out in order to mine anomalies present in the data set using Bayesian networks.

4.3.1 Baseline methods for anomaly detection

We performed experiments over three alternatives for anomaly detection besides COM approach.

1. We used extension of Latent Dirichlet Allocation (LDA) as a baseline method in mining anomalies against our Bayesian approach since both are causal approaches. In Figure 4.5 shows the graphical model for LDA, where plate represents a replication, unshaded circles represent latent variable, shaded circle represents observation, and arrows represent dependency among the variables. LDA is a generative probabilistic model proposed by [14] for modeling text corpora. In LDA, the text corpus is modeled as a collection of M documents, where each document is a set of N_m words. LDA uses a *bag-of-word* assumption, which means the order of words is not important for LDA. Each document is represented by a finite random mixture over latent topics and each latent topic is represented by a distribution over words. The generative process of LDA is described as follows. Let $\mathscr{D}ir(\alpha)$ denotes the Dirichlet distribution with parameter α. Let $\mathscr{M}(\theta)$ denotes the multinomial distribution with parameter θ. The topic proportion of each document, θ_m, is generated from the $\mathscr{D}ir(\alpha)$ ($\theta_m \sim \mathscr{D}ir(\alpha)$). Then LDA

generates $Z_{mn} \sim \mathscr{M}(\theta_m)$ to determine which topic is active out of K topics to later generate the word W_{mn} ($P(W_{mn}|Z_{mn},\beta)$). For each document, the joint distribution of a topic proportion θ, a set of topic assignments for N words z, given the parameters α and β is as follows:

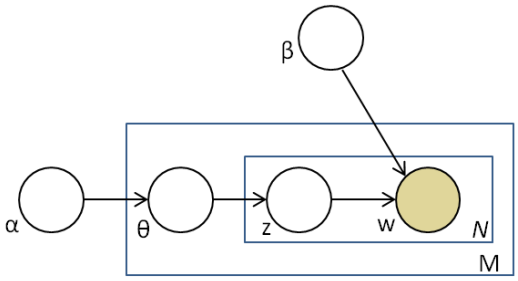


Figure 4.5: Graphical model of LDA

$$P(\theta,z,w|\alpha,\beta) = P(\theta|\alpha)\prod_{n=1}^{N} P(z_n|\theta)P(w_n|z_n,\beta) \tag{4.10}$$

We use several analogies to fit the LDA model: (1) a set of unique feature values as *a set of vocabularies*, (2) each feature value that exists in a domain as *word*, and (3) a category (e.g. normal, abnormal) as a *topic*. With these assumptions, we aim mainly to find the words (features/attributes) that have low probability to appear in a specific topic (category/class), which then we refer as ***"patterns"***. Once such patterns are mined, test cases are checked for anomalies as done in COM approach.

There are several work on outlier detection using the extension of Latent Dirichlet Allocation. Xiong et al. [76] proposed models that extend LDA to mine group outliers from astronomical data set. Ferragut et al. [30] proposed a probabilistic model based on LDA to detect anomalies in data network traffic. For the data sets, they used Internet Protocol (IP) addresses and port information. These data sets were collected for all connections crossing the Oak Ridge National Laboratory network perimeter. Our work is different from theirs as we do not focus on the generative model to detect outliers, but we use the Bayesian network to extract the causal anomalies.

2. k^{th}-NN (k^{th} nearest neighbour) [60] outlier detection technique. In order to make it applicable for a training/testing setting, we changed the method sightly. Instead

of scorning a test point on the basis of its deviation from all other data points belonging to the test data, we scored a data point to extent of which it deviates from the training set. Given a training data set, in order to use the 5^{th} to determine the degree to which a test point is anomalous, we simply use the distance from the point to its 5^{th} in the training set. *A larger distance indicates a more anomalous point.*

3. Local outlier Factor (LOF) anomaly detection [17]. LOF was also modified in the similar way as k^{th}-NN. The LOF of each test point is computed with respect to the training data set in order to score the point.

4.3.2 Experimental setup

For the COMBN algorithm to work, Bayesian network was needed to disseminate knowledge of the domain from which DSAPs can be mined. For this task, we used web based Bayesian modeling software called B-Course [6] to reveal causal relationships among attributes using a data set. Later, the trained Bayesian networks were used in Netica [24] and the COMBN algorithm was developed using Netica Java API.

The procedure to designing the training & testing sets are as follows:

- Based on the class labels given in the data set, we grouped the instances. Instances belonging to one class were grouped under G_1 whereas, all other instances were assembled under group name G_2.
- We randomly took 80% instances each from groups G_1 and G_2 to form $trainingSetG_1$ and $trainingSetG_2$[1] respectively. The rest 20% instances left in groups G_1 and G_2 formed $testSetG_1$ and $testSetG_2$.
- Bayesian network was trained using $trainingSetG_1$.
- As an input for LDA training, both $trainingSetG_1$ and $trainingSetG_2$ were given. This implies LDA model was given instances belonging to two different classes where true class labels were hidden from LDA. We set parameter $K^* = 2$ for LDA learning. The goal of LDA was to learn distribution of features for each group name. After we get the features distribution for each group, we name the

[1]We keep a record of the class label of each instance, but do not include them in $trainingSetG_1$ and $trainingSetG_2$

respective group as same with the class label which has the most frequent features appear in that topic by consulting original data set[2].

- We define anomalies are those test cases which belong to the group G_2, i.e., class on which Bayesian network was not trained. Therefore, ideally, for high accuracy and recall, anomaly detection techniques should discover those test cases which belong to the class other than one used to train Bayesian network.

- For evaluating COMBN and LDA approaches, we used test set = $testSetG_1 \cup testSetG_2$.

- For evaluating k^{th}-NN and LOF, we used test set* = $trainingSetG_1 \cup testSetG_1 \cup testSetG_2$.

The above mentioned steps are also summarized in Figure 4.6.

One reason of designing such training/testing environment was to show that causality matters in mining true outliers. Data points belonging to distinct classes may encode different causal semantic among attributes. For example, a causal dependency, $X_1 \rightarrow X_2$, in the certain class may appear as $X_2 \rightarrow X_1$ in some another class with different probabilistic arrangement. From now onwards, we use the term training set for the data on which models were learnt and, test set as one used for evaluation.

4.3.3 Bayesian networks and data sets

We performed experiments on six real data sets taken from UCI repository [4] to show the credibility of our approach. Besides evaluating our approach on real data sets, we also experimented on well-known Bayesian models taken from network repository maintained by Netica [24] and GeNIe & Smile [33]. For the six real data sets, we designed the training and testing framework as discussed previously. However, the Bayesian models which were taken from the network repository were not learnt since they were pre-defined available on repositories. Also, for these BNs test data were not available to test the accuracy of our algorithm. However, we mined DSAPs from these given Bayesian models in order to show performance of our algorithm on higher dimensions.

[2]By taking this step we found low probable patterns for each class labels

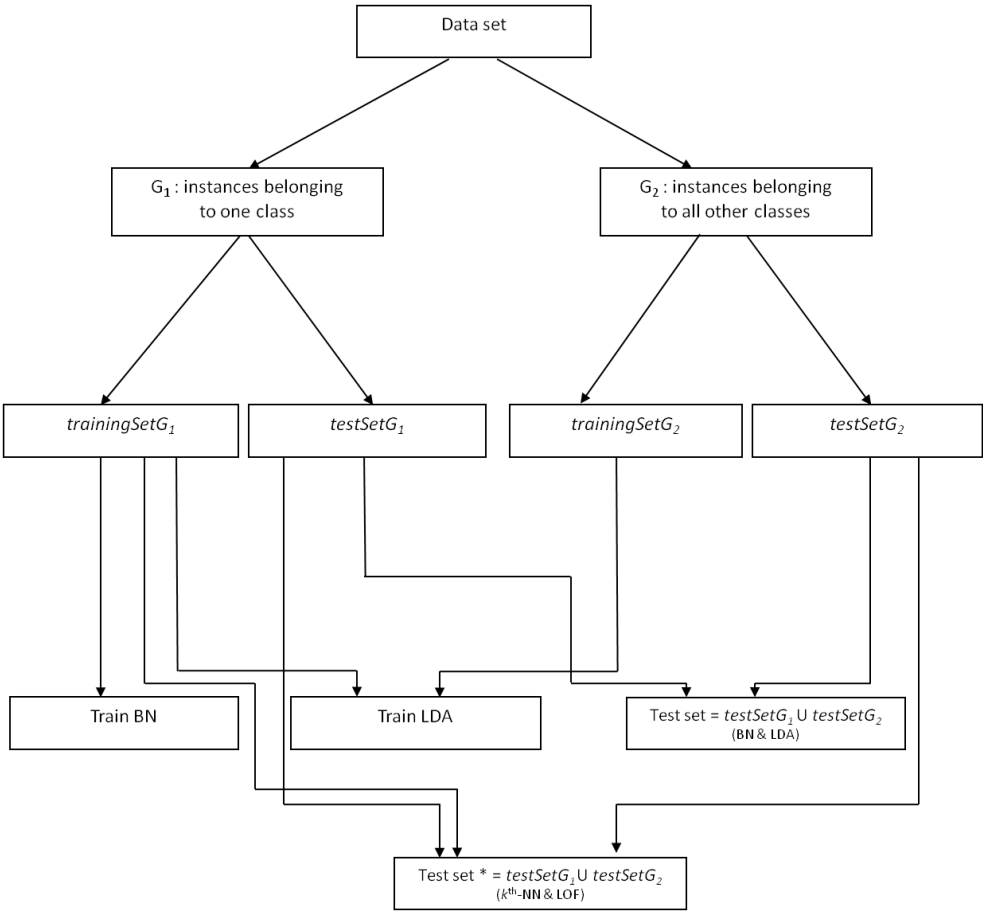


Figure 4.6: Experimental setup

We present the information on pre-defined Bayesian networks and Bayesian networks learnt over six different real life data sets in Table 4.2. The learnt BNs were given same names as the name of data sets available on respective websites to avoid confusion. Column 1 of the table list names of the pre-defined Bayesian networks and number of nodes present in these networks are presented in column 2 of the same table. Column 3 of the table list names of six real data sets for which Bayesian networks were learnt. For these data sets, the classes on which BNs were learnt are presented under the column *Class* (column 4). Next to this column presents information on total number of attributes originally exists in the data set.

4.3.4 Results

We first present Bayesian networks learnt over few data sets, and few pre-defined Bayesian models. Then, results achieved on various data sets are reported. We also

Pre-defined BN		BN learnt		
BN	**Nodes**	**Data set**	**Class**	**Features**
ChestClinic	8	Zoo	Mammal	18
Diabetes learned	9	Mushroom	Eatable	22
Alarm	37	Lymphography	Metastases	18
Win95pts	76	Statlog	Good	20
Pathfinder	135	Congressional Voting Record	Democrat	16
Munin1	189	KDD Cup	Normal	41
Diabetes	413	-	-	-

Table 4.2: Summary of Bayesian networks and data sets

present few subspaces discovered from KDD Cup data set by our algorithm COMBN which explains why the discovered data point is anomaly. Thereafter, we present a detailed analysis on a distance based technique and LDA approach to show their search methodology for mining anomalies and how they are different from COM.

4.3.4.1 Bayesian networks learnt

In Figures 4.7, 4.8, 4.9 and 4.10 we show Bayesian networks learnt using B-course software [6] over four data sets namely, Zoo, Lymphography, Statlog and KDD Cup. In addition to this, two pre-defined Bayesian networks named Chestclinic and Diabetes learned are shown in Figure 4.11 and Figure 4.12 respectively. Names given to nodes encoded in all Bayesian networks which were learnt are same as defined in respective data sets. In Appendix A, we give description on these variables.

4.3.4.2 Experimental evaluation

Table 4.3 summarizes information of all Bayesian networks (learnt and pre-defined) such as number of connected nodes in the model (column 2), total number of links (column 3), conditional probability tables (CPTs) (column 4), total number of DSAPs extracted (column 5) and time taken by COMBN in mining anomalous patterns (column 6). For Bayesian networks which were pre-defined, we deleted function and utility nodes present in the model before using them in the COMBN algorithm. The resulted number of nodes after this operation is shown in column 2 of the table. All continuous variables present in the data sets were categorized to five discrete levels. As discussed before, the computational complexity of our algorithm depends on factors such as number of nodes, links and conditional probability entries present in the Bayesian network.

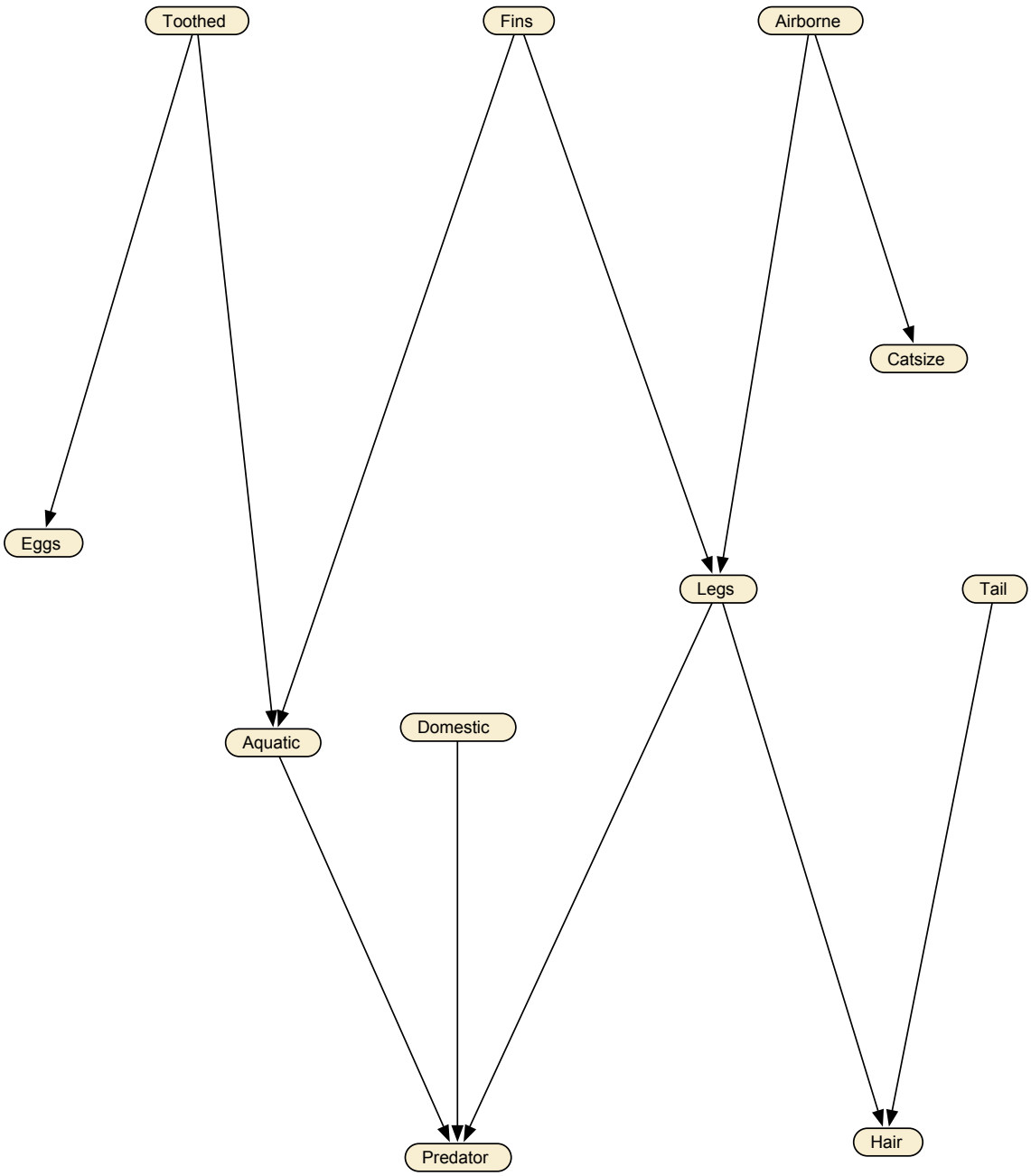


Figure 4.7: Bayesian network on Zoo data set

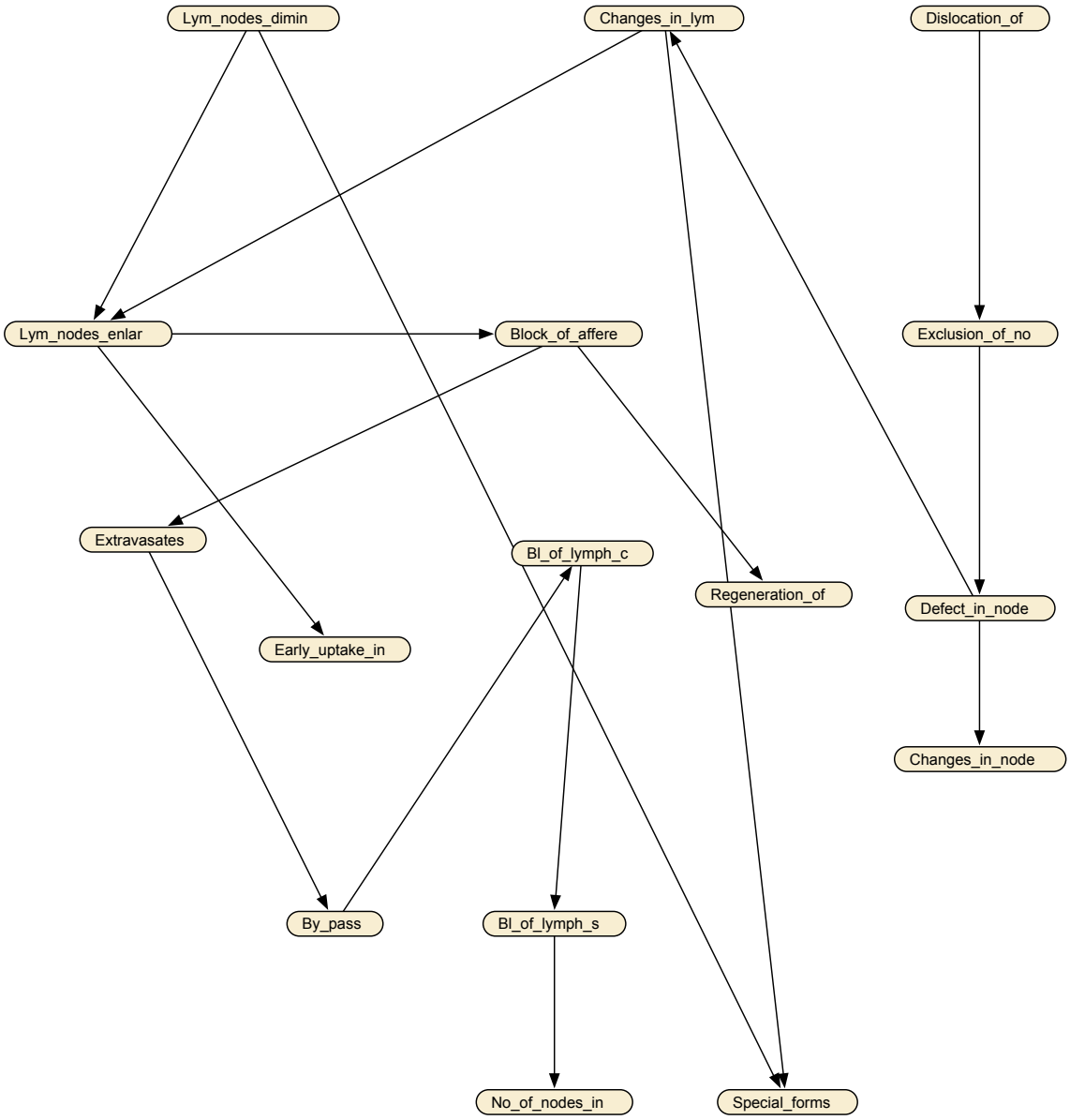


Figure 4.8: Bayesian network on Lymphography data set

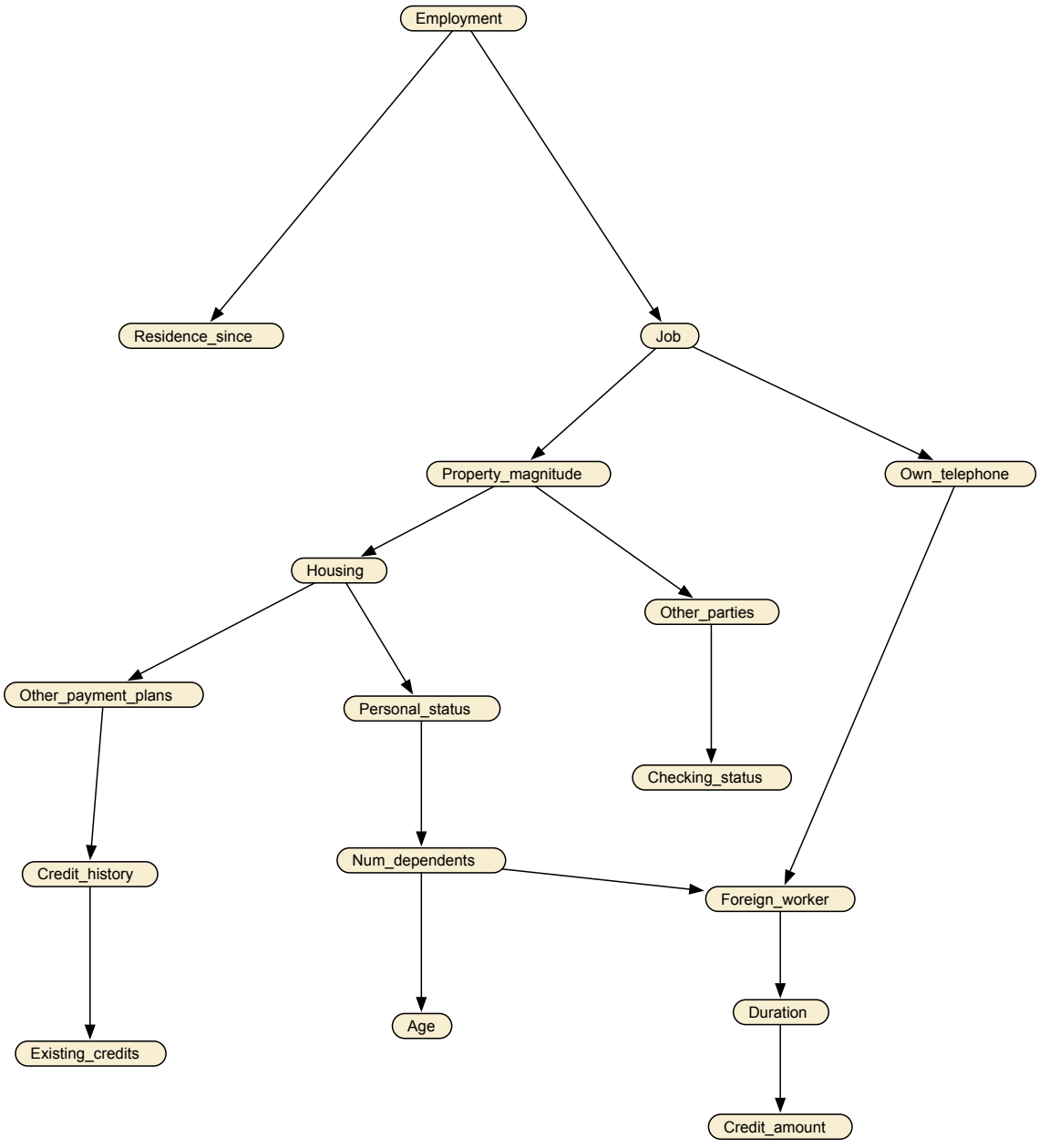


Figure 4.9: Bayesian network on Statlog data set

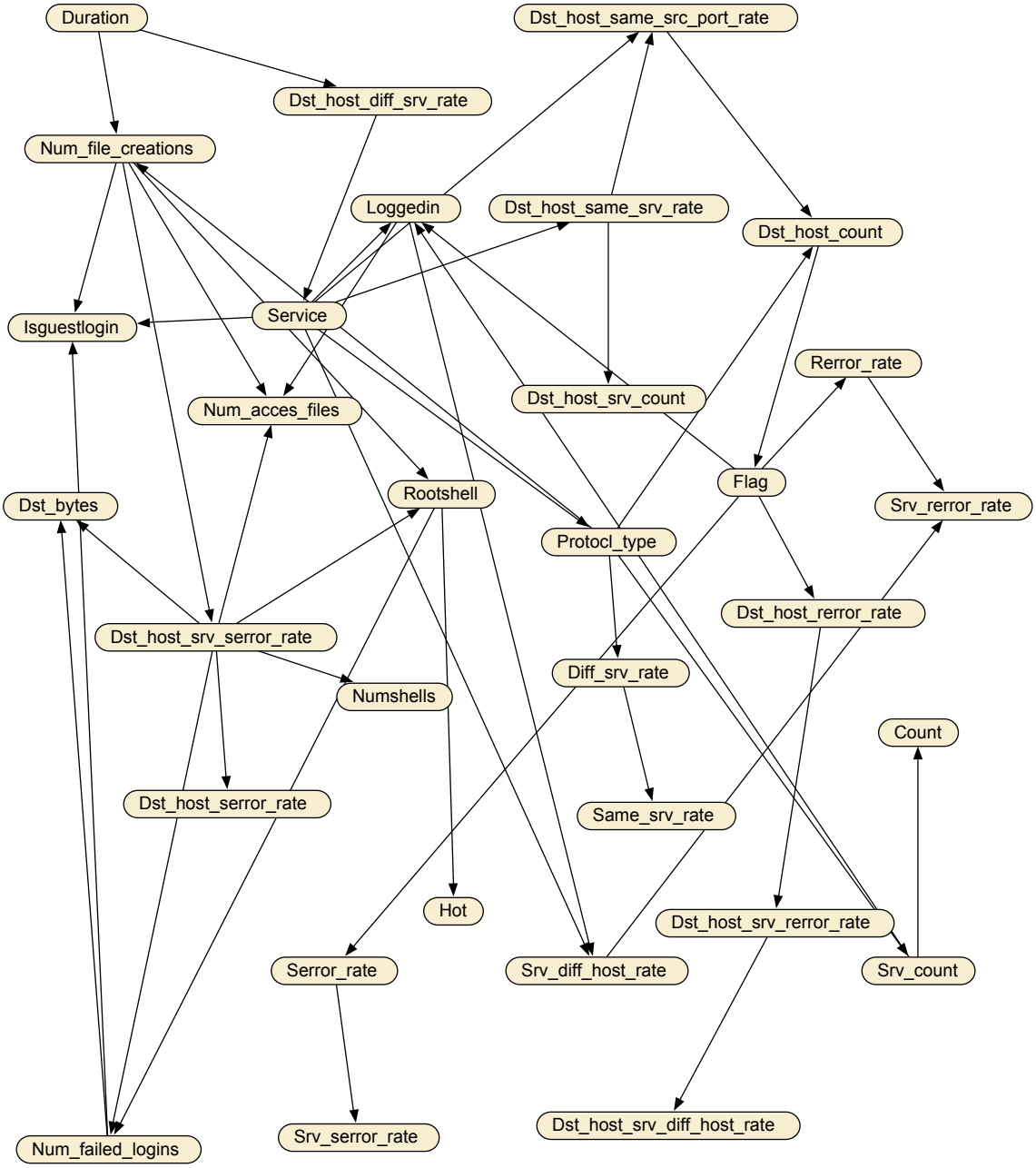


Figure 4.10: Bayesian network on KDD Cup data set

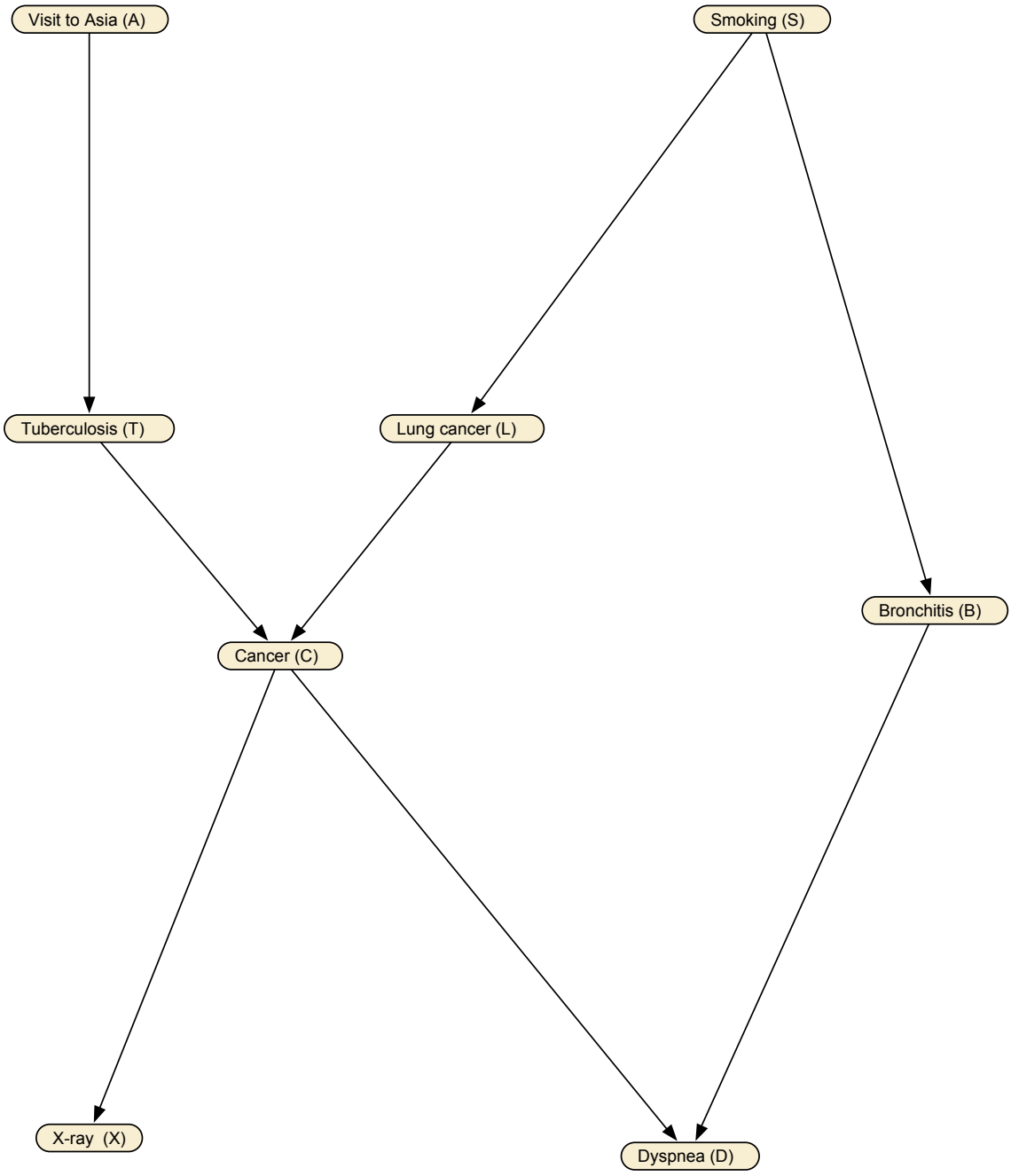


Figure 4.11: Bayesian network: ChestClinic

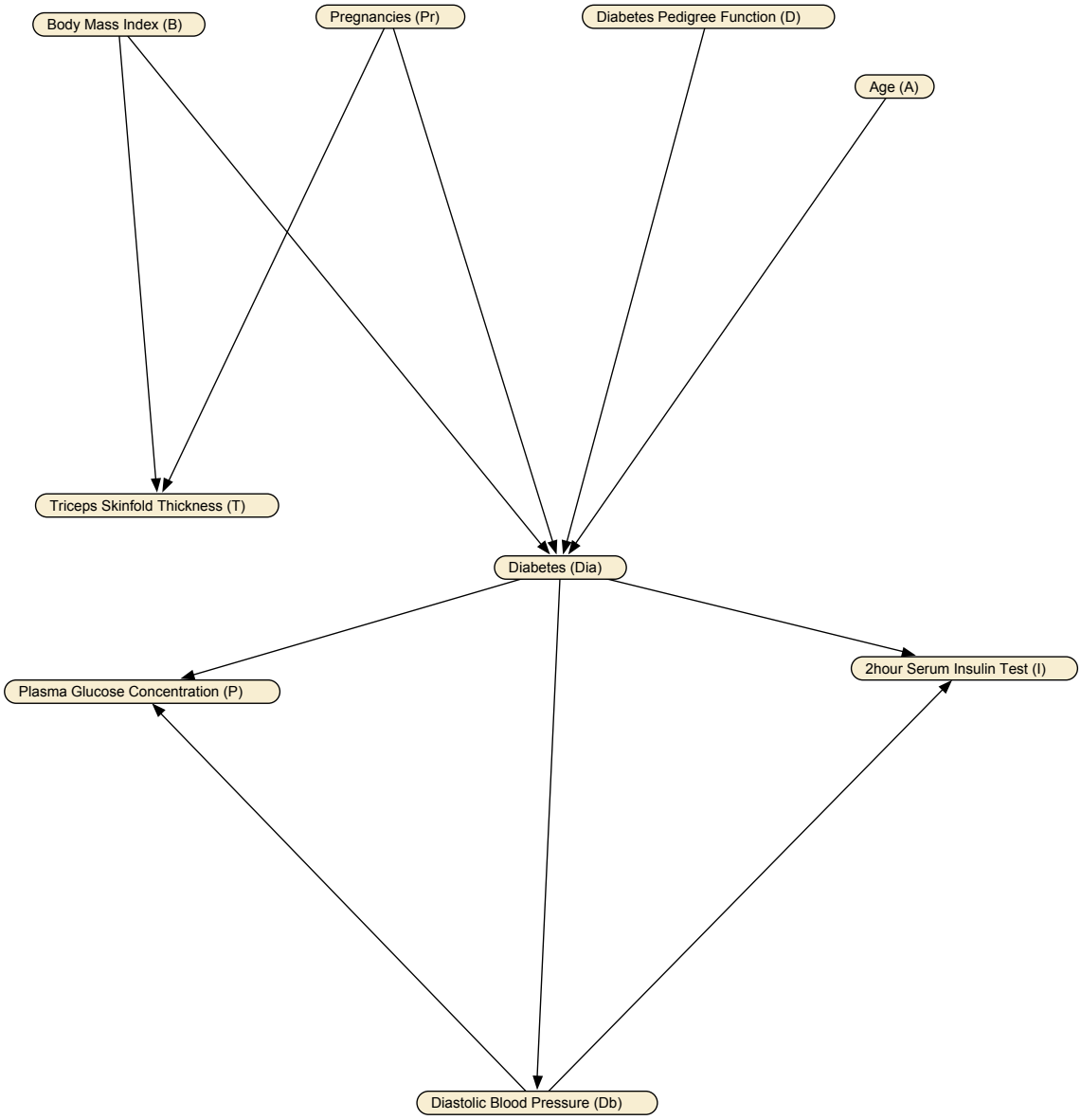


Figure 4.12: Bayesian network: Diabetes learned

BNs	Nodes	Links	CPTs	|DSAPs|	Time in secs.
Zoo	11	11	74	5	1
Mushroom	20	66	20155	71	2
Lymphography	18	16	144	156	1
Statlog	17	17	315	37	1
Congressional Voting Record	16	15	62	7	1
KDD Cup	32	45	29786	277	4
ChestClinic	8	8	36	4	1
Diabetes learned	9	11	159	6	1
Alarm	37	45	752	50	1
Win95pts	75	110	1144	44	3
Pathfinder	100	172	94134	237	16
Munin	156	128	12348	128	26
Diabetes	294	314	312756	1518	210

Table 4.3: Summary: number of nodes, links, CPT entries, number of DSAPs extracted and time taken by COMBN in seconds

Considering these parameters, our algorithm performed well on all data sets. For example, Bayesian network on KDD cup data set of 32 nodes, 45 link and 29786 conditional probability entries, took 4 seconds in mining 277 patterns. Whereas, BN named Diabetes, having huge conditional probability entries of 312,756 took reasonable time of 210 seconds. Table 4.4 summarizes information on data sets used in experiments and compares the quality of precision/recall obtained using COMBN, LDA, k^{th}-NN and LOF anomaly detection techniques. Column 1 of the table lists data set names. Precision and recall on four outlier detection approaches are presented in columns 2 and 3 respectively.

We set parameter $\tau = 50\%$, $minconf = 10\%$ and $maxconf = 90\%$ in the algorithm COMBN. For a reasonable comparison between our approach and LDA, we took the top n low probability patterns mined by LDA where, n was equal to $(\tau \times |DSAPs|)$ set in algorithm COMBN. In k^{th}-NN approach, we set $k = 5$ for experiments. We obtained encouraging results by our algorithm with precision and recall more than 70% for almost every data set. Our approach performed very well on a real life network intrusion data set (KDD CUP) giving precision of 96% and recall of 99%. There were 22 different attack types present in this data set grouped under four categories namely, Denial Of Service (DOS), User to Root (U2R), Probe and Remote to Local (R2L). Out

Data set	**Precision** (**COMBN, LDA, k^{th}-NN, LOF**)	**Recall** (**COMBN , LDA, k^{th}-NN, LOF**)
Zoo	(**.91**, .69, .62, .56)	(**.99**, 1, .62, .52)
Mushroom	(.62, .56, .52, .66)	(.71, 1, .62, .61)
Lymp.	(.72, .50, .69, .57)	(.83, 1, .69, .66)
Statlog	(**.86**, .49, .52 .49)	(**.77**, .49, .49, .45)
Cong. V. R.	(**.91**, .59, .54, .45)	(**.95**, .94, .64, .48)
KDD Cup	(**.96**, -, .72, .41)	(**.99**, -, .66, .44)

Table 4.4: Summary: precision and recall achieved using COMBN, LDA, k^{th}-NN and LOF algorithms

of 22 attacks present in the data set, our technique, failed only in discovering two attacks namely, *Buffer_overflow*, *Guess_pwd* (there were zero samples for attacks *Warezmaster* and *Imap* in the data set).

The LDA result for KDD Cup is not shown because this data set contained imbalance proportion of instances belonging to each class (normal and 22 different attack types). In order to mine the patterns of low probability for each class type, it was important to train LDA model on these classes. However, we formed a ten randomized data set from original KDD Cup data set addressed as KDD Cup* especially for LDA in order to see its performance on a real network intrusion detection data set. Each randomized KDD Cup* data set was formed by taking three steps. In step 1, we chose instances belonging to class labels: normal and, six distinct attacks namely *Neptune*, *Smurf*, *Satan*, *Back*, *Warezclient* and *Teardrop*. In step 2, out of chosen seven classes we randomly chose m number of instances from each group. We took equal number of instances so as to provide LDA with a fair learning environment. Finally in step 3, 80% of m instances under each class were randomly selected to form a training set and rest 20% formed the test set. Each random sample of data set KDD Cup* contained 320 anomalies (or attacks) and 32 normal instances.

Results achieved on KDD Cup* using LDA were not encouraging. The average true positive rate achieved was 320.0 and false positive rate of 32.0. The result implies that, LDA approach was not able to distinguish between anomalies and normal instances. However, in comparison COM achieved average true positive rate of 290.6 and false positive rate of 3.8. Interestingly, being trained on same training set of normal instances and using similar methodology of testing test sets for anomalous patterns, our approach worked reasonably good as compared to LDA. We found this result very interesting and

investigated further on why false positives were so high for LDA in comparison to our approach in order reveal discrepancy in the results. We discuss this in Section 4.3.7.

We mentioned previously that one of the key advantage of our technique is that it can both identify anomalies and explain the reason of their anomalous nature. In Table 4.5 we show the subspaces which were targeted by various attack types present in KDD Cup data set. For example Table 4.5 reveals the fact that *Smurf* (attack type) aim for subspace,($Diff\_srv\_rate$[0-0.1] $\rightarrow$ $Same\_srv\_rate$[>0.9]).

We discussed in Section 4.2 that if the number of DSAPs extracted is large for a data set of low dimensionality then, it may lead to high false positive rate because of multiple hypothesis problem. In order to show the relation among the number of attributes, numbers of DSAPs extracted and false positive rate, first refer Fig. 4.13a. It shows an increase in TP and FP with an increase in percentage of number of DSAPs extracted for Statog data set. We set the parameter τ from 10% to 100% and calculated TP and FP at every scale of parameter τ. The number of normal and anomalous data points was 140 and 300 respectively in the test set of this data set. TP is represented by a thick line, whereas FP is shown by a dashed line. The graph clearly shows an increase in TP with an increase in percentage of total DSAPs until τ = 70%. After this point, there is not much change in TP. However, there is always a constant increase in FP till τ = 100%. This explains the fact that if we consider all DSAPs extracted to discover anomalies then, we may end up having good recall but poor precision. On the other hand, if we consider only top few low scored interesting DSAPs then; we can get both good precision and recall.

We also show in Figure 4.13b that the patterns of TP and FP for the same data set on similar scale of τ, but this time DSAPs were ranked in a descending order. Here the trend of TP until τ = 50% is increasing at a very low pace. This indicates that top high scored DSAPs were least interesting from the anomaly discovery perspective. Interestingly, TP grows right after τ = 50% which clearly shows the contribution of low scored DSAPs in mining true anomalies. On the similar bases, we present in Fig. 4.14a and 4.18b the trend of TP and FP on τ parameter achieved on KDD Cup data set, where sensitivity score was sorted in ascending and descending orders respectively.

As discussed in Section 4.2, mining of DSAPs from the given data set is dependent on two parameters namely, *minconf* and *maxconf*. Both of these parameters are independent of each other. Parameter *maxconf* is used by $\mathbf{R}_1$ while, $\mathbf{R}_2$ depends on *minconf*. In Figure 4.15a, we show, if only $\mathbf{R}_1$ is applied on Mushroom data set, then

Attack	DSAPs
Neptune	1. *Num_file_creations*[0-2.8] → *Dst_host_srv_serror_rate*[0-0.1] 2. *Duration*[0-5832.9] → *Dst_host_diff_srv_rate*[0-0.1]
Back	1. *Flag*[SF] → *Dst_host_rerror_rate*[0.1-0.2] 2. *Dst_host_same_srv_rate*[>0.9] → *Dst_host_srv_count*[178.5-204]
Warezclient	1. *Rootshell*[0-0.1] → *Hot*[3-6] 2. *Dst_host_srv_serror_rate*[0.4-0.5] → *Dst_host_serror_rate*[0.3-0.4]
Pod	1. *Num_file_creations*[0-2.8] → *Dst_host_srv_serror_rate*[0-0.1] 2. *Duration*[0-5832.9] → *Dst_host_diff_srv_rate*[0.2-0.3]
Teardrop	1. *Duration*[0-5832.9] → *Dst_host_diff_srv_rate*[0.6-0.7] 2. *Duration*[0-5832.9] → *Dst_host_diff_srv_rate*[0.5-0.6]
Portsweep	1. *Srv_count*[0-51] → *Count*[0-51.1] 2. *Protocol_type*[tcp], *Dst_host_same_src_port_rate*[>0.9] → *Dst_host_count*[51-76.5]
nmap	1. *Num_file_creations*[0-2.8] → *Dst_host_srv_serror_rate*[0-0.1] 2. *Dst_host_srv_rerror_rate*[0.2-0.3] → *Dst_host_srv_diff_host_rate*[0.8-0.9]
Ipsweep	1. *Protocol_type*[tcp], *Dst_host_same_src_port_rate*[>0.9] → *Dst_host_count*[51-76.5] 2. *Dst_host_diff_srv_rate*[0.7-0.8] → *Service*[ecri]
Satan	1. *Dst_host_srv_serror_rate*[0.4-0.5] → *Dst_host_serror_rate*[0.7-0.8] 2. *Protocol_type*[tcp] → *Diff_srv_rate*[0.4-0.5]
Smurf	1. *Diff_srv_rate*[0-0.1] → *Same_srv_rate*[>0.9] 2. *Duration*[0-5832.9] → *Dst_host_diff_srv_rate*[0-0.1]
Land	1. *Service*[http], *Loggedin*[0] → *Srv_diff_host_rate*[0-0.1] 2. *Protocol_type*[tcp] → *Srv_count*[0-51]
Spy	1. *Dst_host_count*[204-229] → *Flag*[SF] 2. *Flag*[SF] → *Rerror_rate*[0-0.1]
Perl	1. *Service*[http], *Loggedin*[0] → *Srv_diff_host_rate*[0.3-0.4] 2. *Srv_count*[0-51] → *Count*[408-459]
Phf	1. *Dst_host_count*[0-25.5] → Flag[RSTR] 2. *Protocol_type*[tcp] → *Srv_count*[0-51]
Multihop	1. *Service*[http], *Loggedin*[0] → *Srv_diff_host_rate*[0.3-0.4] 2. *Srv_count*[0-51] → *Count*[408-459]
Ftp_write	1. *Rootshell*[0-0.1] → *Hot*[3-6] 2. *Flag*[SF] → *Rerror_rate*[0-0.1]
Rootkit	1. *Flag*[SF] → *Rerror_rate*[0.2-0.3] 2. *Flag*[SF] → *Rerror_rate*[0-0.1]
Loadmodule	1. *Diff_srv_rate*[0-0.1]→ *Same_srv_rate*[0.4-0.5] 2. *Protocol_type*[tcp] → *Srv_count*[0-51]

Table 4.5: Domain specific anomalous causal subspaces discovered using COMBN algorithm for various attacks present in KDD Cup data set

how number of DSAPs decreases on increase of *maxconf* parameter. Similarly, in Figure 4.15b, we show application of $\mathbf{R}_2$ on the same data set using parameter *minconf*. Results show that a small variation in *minconf* and *maxconf* did not effect discovery of DSAPs.

4.3.5 Robustness of Rules $\mathbf{R}_1$ and $\mathbf{R}_2$

We performed a test in order to evaluate robustness of rules $\mathbf{R}_1$ and $\mathbf{R}_2$. Our goal through these tests is to show that rules we propose helps mining "interestingly rare patterns" than mining "irrelevant patterns or noise".

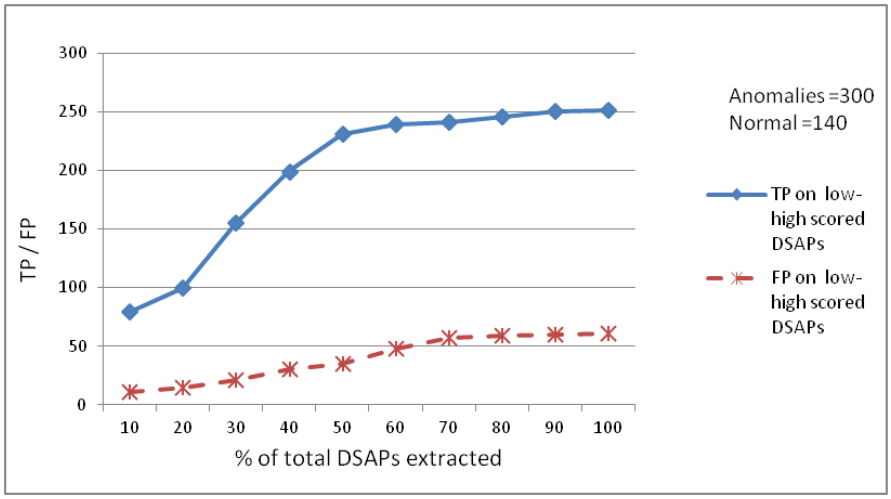
The test we used is based on concept of joint probability distribution (JPD), refer Equation 4.11 in BN. Joint probability distribution in BN is a product of priors and conditional probability across each of the variable in a BN.

$$P(X_1, X_2...., X_{|X|}) = \prod_{i=1}^{|\mathbf{X}|} P(X_i|Pa(X_i)) \tag{4.11}$$

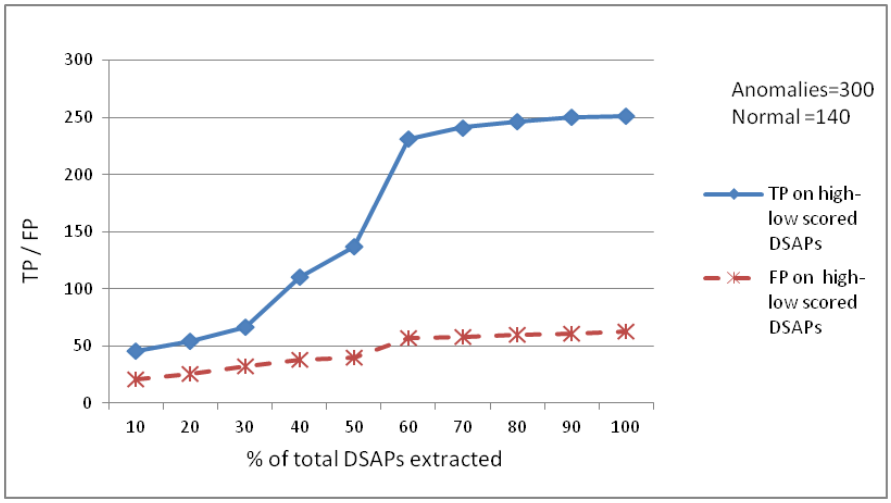
An important observation here is, product of priors and conditional probabilities, which constitute a score of a given test case, can give rise to four different situations namely,

1. low prior and high conditional probability
2. high prior and low conditional probability
3. low prior and low conditional probability
4. high prior and high conditional probability

A joint probability actually is a product of the above four factors or we can say joint probability is formed by the combination of above listed situations. However, it is always possible that any situation occur any number of times, while at the same time it is not also necessary that every situation will be present in the product. This depends upon values taken by attributes and their structure of relationship. Of the four situations, the situations listed at one and two are the only case where there is a conflict between the evidence and event conditional probability provides for a theory and our prior belief about the plausibility of that theory and hence an indication of potential outlying situations. Griffiths and Tenenbaum [35] defines situations one and two above

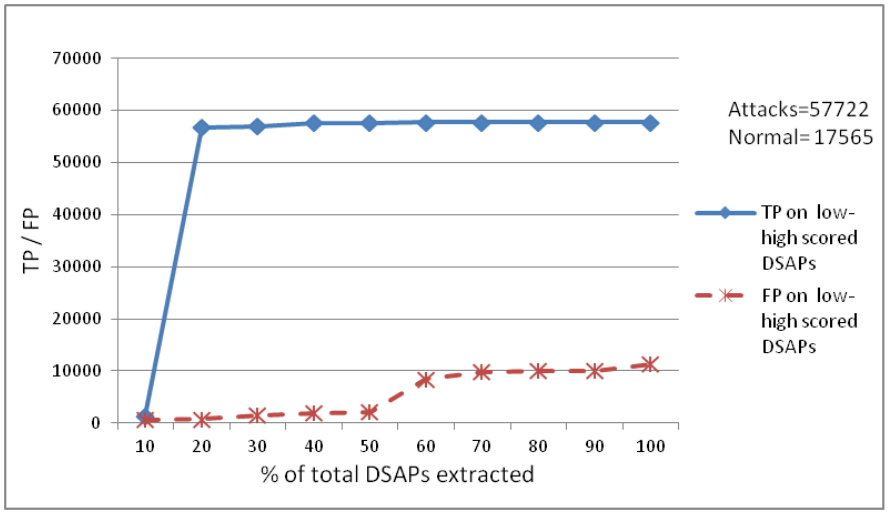


(a) For Statlog data set pattern of TP and FP achieved on parameter τ scaled from 10% to 100% when DSAPs sorted in ascending order

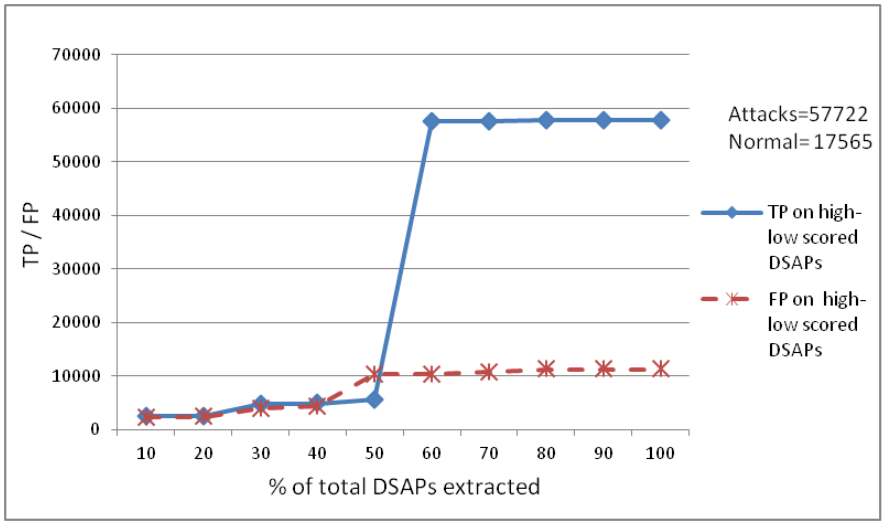


(b) For Statlog data set pattern of TP and FP achieved on parameter τ scaled from 10% to 100% when DSAPs sorted in descending order

Figure 4.13: Performance of COMBN on τ parameter in Statlog data set

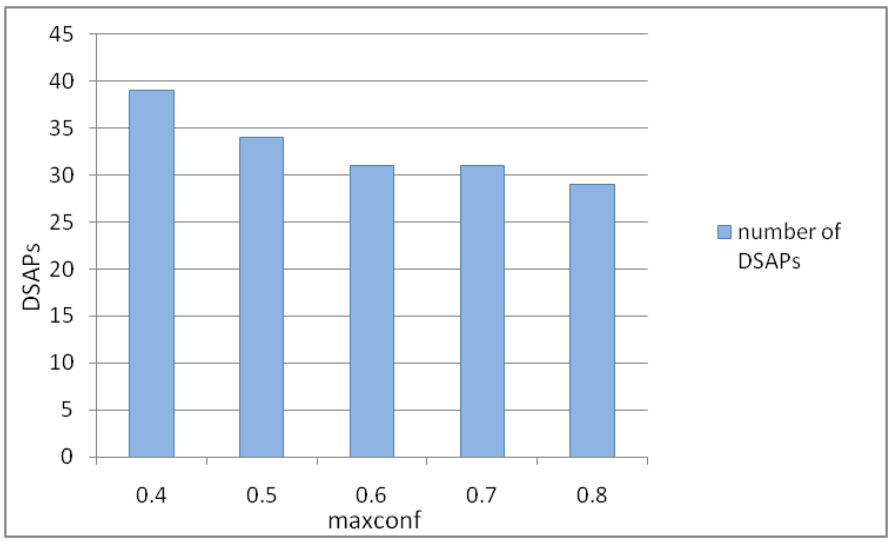


(a) For KDD Cup data set pattern of TP and FP achieved on parameter τ scaled from 10% to 100% when DSAPs sorted in ascending order

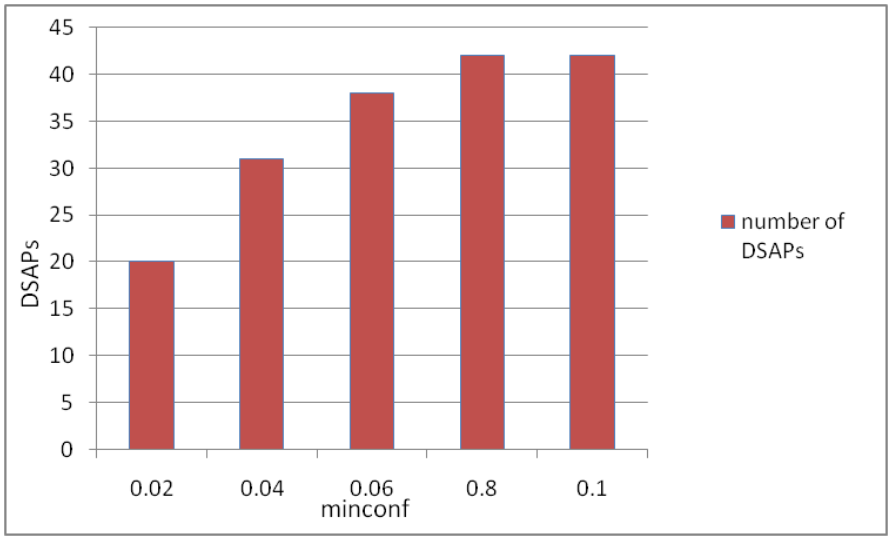


(b) For KDD Cup data set pattern of TP and FP achieved on parameter τ scaled from 10% to 100% when DSAPs sorted in descending order

Figure 4.14: Performance of COMBN on τ parameter in KDD Cup data set



(a) Impact of parameter *maxconf* on discovery of domain specific anomalous patterns (DSAPs)



(b) Impact of parameter *minconf* on discovery of domain specific anomalous patterns (DSAPs)

Figure 4.15: Impact of parameters *maxconf* and *minconf* on discovery of domain specific anomalous patterns (DSAPs)

as mere and suspicious coincidence respectively. From outlier mining point of view, low unconditional probability is most likely a "noise event" unless there exists a variable for which there is high conditional probability. Situation three is example of noise. High support and high confidence is example of high correlation and association among attributes. The focus of association rule mining is to discover such patterns from data.

Logically a joint probability of a test case will be high which has maximum number of fourth situation listed above. Contrary to this, joint probability of test case will be low which has maximum number of first three situations listed above. This implies, if we are to test a number of data points with the objective of mining top n anomalies from it then, JPD in BN can be used to score each instance. However, interestingly enough, the independent factors forming JPD in BN, i.e., P(X | Pa(X)) are effectively the patterns which rules $\mathbf{R}_1$ (situation one) and $\mathbf{R}_2$ (situation two) aim for subject to two conditions mentioned in Equations 4.8 and 4.9 respectively.

To test if the above discussed theory exists and our rules are genuine, top n data points with low JPD computed using BN should carry patterns mined using our probabilistic rules $\mathbf{R}_1$ and $\mathbf{R}_2$. We perform extensive experiments on data sets to confirm this. As a result we found that for each data set, patterns discovered using $\mathbf{R}_1$ and $\mathbf{R}_2$ appeared with more than average of 90% probability in every top n low scored data point in terms of JPD. An obvious question here is: why do we need approach of probabilistic rules when concept of JPD can give us equally good results? It is because of two reasons: (1) application of rules are not only useful in mining outliers, but they also eventually help provide justification on why the data point is discovered as anomaly. Using JPD concept, we can only find top n outliers. Additional contextual knowledge on discovered anomalies cannot be provided using JPD theory and, (2) Using JPD theory, we are first dependent in finding joint probability of each data point and later sorting them in ascending order to reveal top n outliers. However, probabilistic rules gives us freedom from such dependency and sorting. A test point satisfying either rule $\mathbf{R}_1$ or $\mathbf{R}_2$ will be identified as an anomaly.

4.3.6 Relevance of COM Methodology

Our emphasis in this section is on the usefulness and relevance of our approach in discovering genuinely anomalous patterns. Any outlier detection technique is novel if it can validate anomalous behavior of the observations and can provide insights into the

fact as to why these observations are suspicious. Such insights not only give understanding on data but helps in improving knowledge of the domain. The most authentic way to validate outliers discovered by any outlier detection technique is by evaluating observations using domain knowledge. However, as expertise of the particular domain is not always readily available to disseminate knowledge about the domain and validate outliers; so we restrict explaining relevance of our approach using a simple yet meaningful Bayesian network on a medical diagnose problem represented in Figure 4.11.

We present relevance and quality of our approach by mining DSAPs from this BN. The idea is, if an explanation of DSAPs discovered by our approach on BN in Figure 4.11 could be justified by the domain and common sense knowledge as an unseen yet interesting knowledge then it could give a strong indication of relevance of our approach. We chose BN named ChestClinic from several available because the relationship among attributes and the general knowledge of the domain is very easy to understand and hence explaining anomalous patterns our approach will mine. Following are the four DSAPs extracted using rules $\mathbf{R}_1$ or $\mathbf{R}_2$ with parameters *minconf* and *maxconf* set to 10% and 80% respectively.

1. Visit to Asia[visit](1%) $\rightarrow$ Tuberculosis[absent](95%)
2. Smoke[smoker](20%) $\rightarrow$ Lung cancer[absent](90%)
3. Cancer[false](92.6%) $\rightarrow$ X-ray[abnormal](5%)
4. Cancer[false](92.6%), Bronchitis[absent](75%) $\rightarrow$ Dyspnea[present](10%)

We amended the notation of causal subspace defined in Equation 4.7 with additional information which is represented in angular braces. Information in angular braces represents state of the variable. With respect to BN in Figure 4.11, we explain four outlier subspaces identified as follows:

1. Percentage of people who makes visit to Asia(1%) is unlikely to have tuberculosis (95%). This is a suspicious event because we do not have enough evidence(1%, which is very small) to this fact.
2. Referring to second subspace, there is one cause of lung cancer, i.e., smoking. A lay mans opinion says, a person who smokes is mostly likely to get affected by lung cancer. For the given instance, value of the variable smoker is “smoker” and

value of lung cancer is "absent". This obviously indicates a new dimension to knowledge that there could be other causes leading to lung cancer. The support of smoke is 20%, which is considered as *minsupp* because smoke has only states with same probabilities.

3. In the third subspace, intuitively, a person suffering for cancer should have abnormal x-ray. Whereas, for this observation, cancer is absent but still x-ray report is abnormal. It raises question as to why x-ray is abnormal when cancer is absent. This lead us to a new knowledge that abnormal x-ray is not only affected by the presence of cancer but there could exist other factors causing abnormal x-ray.

4. Similarly, for the fourth subspace, two causes of disease dyspnea namely, cancer and bronchitis are absent but still disease dyspnea is present.

4.3.7 Discussion

The performed experiments were designed to answer the following research question: *Is an approach that treats the features: (a) in an independent fashion or, (b) based on the causal semantic, can perform better in discovering the true anomalies?* Based on the results presented in Table 4.4, we may conclude that by considering the causality can lead to substantially good results. However, it was interesting to see why k^{th}NN, LOF and LDA techniques performed poorly in almost all data sets over Bayesian approach. In this section, we present critical analysis on search methodologies of all the outlier detection techniques followed in this chapter to reveal the mismatch in their results.

What COM approach follows?

Our proposed approach based on Bayesian network tightly integrates relationships among features of the domain and plausibility of an event in probabilistic terms. By exploiting relationships, low and high likely events can be interpreted. As discussed in Section 4.3.5, DSAPs extracted by rules $\mathbf{R}_1$ and $\mathbf{R}_2$ also appears in top n low probability data points if we compute JPD of every test case. So, in order to show insights of COM methodology we use concept of JPD in BNs. We explain COM approach inline with theory of JPD using two pre-defined Bayesian models namely ChestClinic (refer Figure 4.11) and Diabetes learned (refer Figure 4.12). For analysis, we simulated data sets using Netica software [24] for these BNs so that top n outliers using JPD can be

computed. We chose Bayesian networks with minimum number of attributes so that analysis through graph can easily be explained. For easy reference we address variable names modeled in these networks by their initials as represented within angular braces next to their names. The JPD for ChestClinic and Diabetes learned Bayesian networks is represented by Equations 4.12 and 4.13.

$$P(A,S,T,L,B,C,X,D) = P(X|C) \times P(D|C,B) \times (C|T,L) \times P(T|A) \times P(L|S) \times P(B|S) \times P(S) \quad (4.12)$$

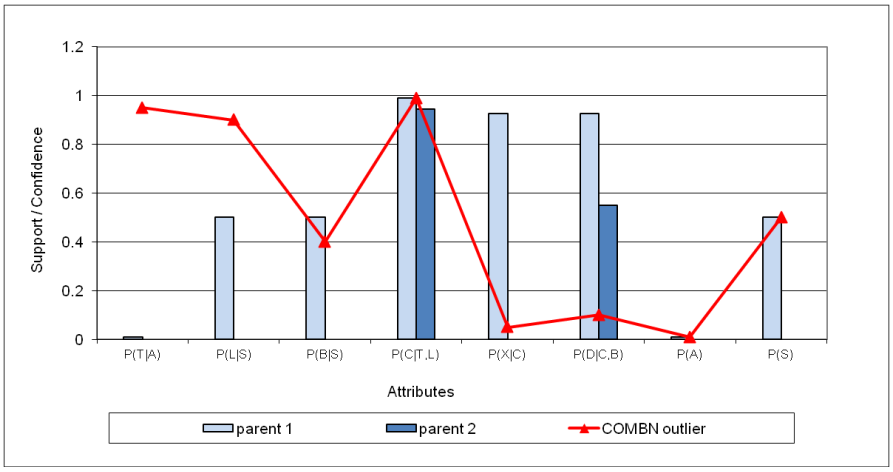
$$P(B,Pr,D,T,Dia,A,P,Db,I) = P(T|B,Pr) \times P(Dia|B,Pr,D,A) \times (P|Dia,Db) \times P(Db|Dia) \times P(I|Dia,Db) \times P(B) \times P(Pr) \times P(D) \times P(A) \quad (4.13)$$

Using simulated data sets from these networks, we computed JPD for every instance and took top n low scored instances to explored inner structure of the data points. It is important to mention here, top n observations were scored low in the Bayesian network because they were having maximum patterns of two qualitative rules. However, by structuring these anomalous instances in individual probabilistic terms we can observe these anomalous patterns and can explain COM approach. Such representation not only indicates search methodology of our technique but also gives understanding on data in general.

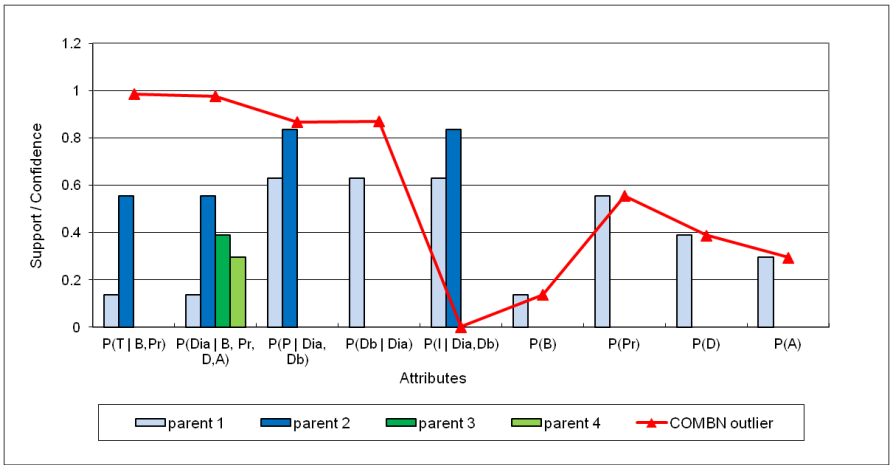
Figures 4.16a and 4.16b represents pattern of top outlier in terms of conditional probabilities (confidence) and prior (support) which together constitutes joint probability in the Bayesian network. Figures 4.16a and 4.16b belong to ChestClinic and Diabetes learned named Bayesian models respectively. The X-axis of the graph represents attributes and Y-axis represents support of parent node in bars and confidence in child node through trend line. Here, support of the parent node is defined using Equation 4.1. For graph in Figure 4.16a, first six attributes are child node whereas, rest two are independent nodes. Referring to Bayesian model named ChestClinic in Figure 4.11 and first bar in graph of the Figure 4.16a indicates, support in parent node Visit to Asia is nearly zero, but confidence in direct child (Tuberculosis) of this parent node (Visit to Asia) is above 95% which is quite high as represented by the point on the trend line just above the bar.

More than one bar at the same position represents number of parents linked with

that child. For example, child node Cancer has two parents (Tuberculosis and Lung cancer) and hence shown by two different support bars in fourth term. Trend shown in Figure 4.16a specifies subspaces which define outlier. Terms first, second, fifth and sixth were uncovered by the qualitative rules. Not only outlying subspaces are visible but normal subspaces can also be interpreted by the observing the graph. Causal subspace, (Tuberculosis, Lung cancer $\rightarrow$ Cancer) is example of high support and high confidence and hence is normal. Similar explanation can be followed for the graph in Figure 4.16b.



(a) Pattern of top COMBN outlier discovered in ChestClinic data set. The bars represent support of the parent attribute(s) and conditional probability in a child node is represented by the trend line. Terms first, second, fifth and sixth were uncovered by the qualitative rules



(b) Pattern of top COMBN outlier discovered in Diabetes learned data set. The bars represent support of the parent attribute(s) and conditional probability in a child node is represented by the trend line. Terms first, second and fifth were uncovered by the qualitative rules

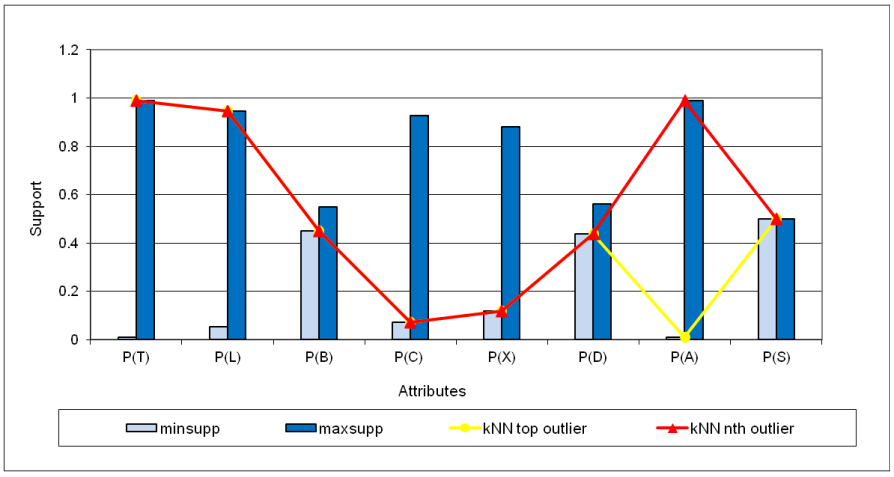
Figure 4.16: Pattern of top COMBN outlier: (a) discovered in ChestClinic data set (b) discovered in Diabetes learned data set

What Nearest neighbor technique follows?

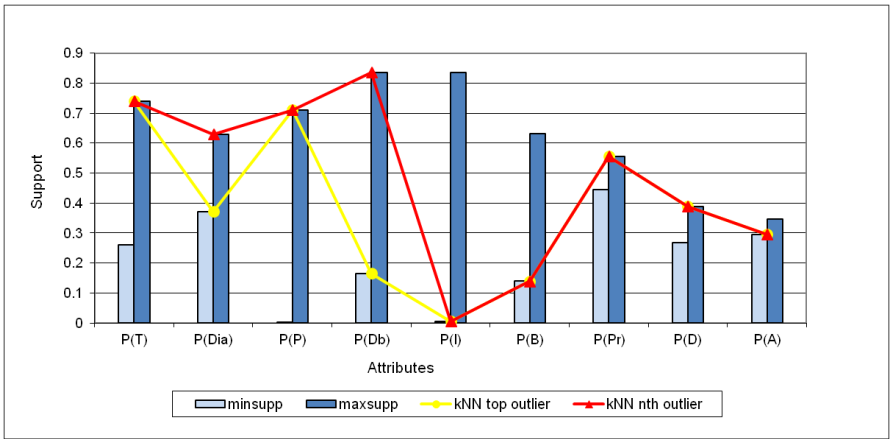
In this section, we address on why there is a mismatch between outliers as observations "which are far away from their neighbors" and "real" outliers as identified using Bayesian approach. The major difference between nearest neighbor and Bayesian approach can be summarized as follows: distance based technique treats every attribute of the domain uniformly whereas, for the Bayesian approach, treatment with attributes depends upon relationship among attributes. Any distance based approach will find a pair wise distance between two objects and will declare an object to an outlier which is far away from k nearest neighbors. Intuitively, it implies that an object declared as an outlier does not have enough support by the nearest neighbors, so, is isolated and far away from the dense area. Contrary to this, a dense cluster is formed by those data points which has similar support from the nearest neighbors which is the reason they satisfy condition of k nearest neighbor and hence are normal. Thus distance based approaches look for those data points where maximum number of attributes have low support. On the other hand, Bayesian approach considers both conditional probability (confidence) and unconditional probability (support) in order to discern between abnormality and normality.

For analysis on nearest neighbor approach, we used simulated data sets from ChestClinic and Diabetes learned BNs to k^{th}-NN approach in order to find top n distance based outliers. The analysis on top and n^{th} outlier discovered on mentioned BNs are shown in Figures 4.17a and Figure 4.17b respectively. In the figures, X-axis represents attributes of the domain and Y-axis represents support of the attributes. Two bars on every attribute of X-axis represents minimum and maximum support attribute has in the Bayesian network. Minimum support of the attribute follows Equation 4.5 and maximum support of an attribute in the Bayesian network is represented by Equation 4.6.

In addition, two trend lines reveal the pattern of top and n^{th} outlier discovered by distance based technique. Interestingly, top outlier has six attributes with low support (indicated by yellow trend line) whereas, for n^{th} outlier, five attributes have low support (indicated by a line) for the ChestClinic data set as represented by the graph in Figure 4.17a. Similar pattern is observed in Figure 4.17b. For few data sets we found, distance based outliers chose those data points as outliers where support of few attribute is near to minimum support if not minimum support exactly.



(a) Pattern of top and n^{th} k^{th}-NN outlier in the ChestClinic data set. The bars represents minimum and maximum support of the attribute in the Bayesian network and two trend lines represents k^{th}-NN top and n^{th} outliers respectively. Top outlier (yellow line) has six attributes with low support whereas, n^{th} outlier (red line) has five attributes with low support



(b) Pattern of top and n^{th} k^{th}-NN outlier in the Diabetes learned data set. Top outlier (yellow line) has five attributes with low support whereas, n^{th} outlier (red line) has three attributes with low support

Figure 4.17: Pattern of top and nth k^{th}-NN outlier: (a) discovered in ChestClinic data set (b) discovered in Diabetes learned data set

Why LDA performed poorly?

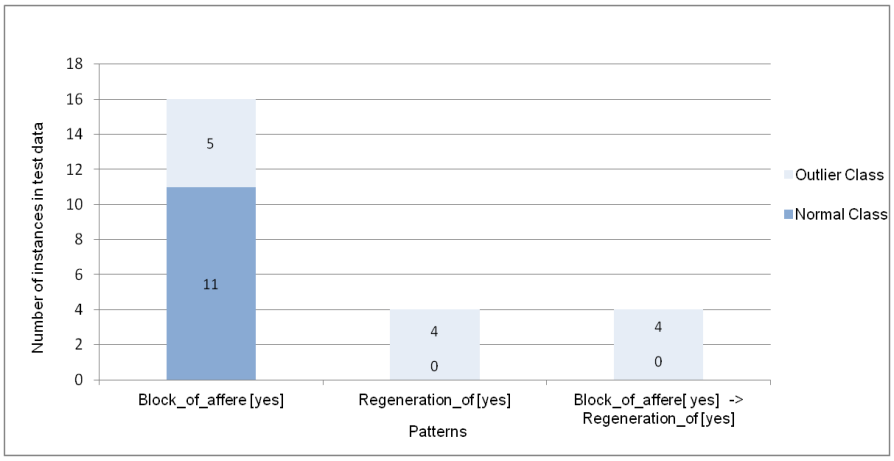
In the LDA framework designed for anomaly detection, from a given number of documents and number of topics, the goal was to assign words appearing in the documents to one of the topic with high probability; where words were considered as values of the features and topic as classes, for example normal and abnormal. This means, LDA model looked for relationship between values of the features and class labels, rather than discovering relationship among values of distinct features appearing in the document. Intuitively, this implies that the features were considered independent of each other. In LDA analogy, topic was constructed by a distribution of words and since topic was also drawn from a Dirichlet distribution so, it was obvious that some words get assigned to have high probability, while others assigned to have low probability. Or in the other words, LDA discovered probability of a word in a topic.

In contrast to LDA, our approach worked in a cause-effect framework captured by the Bayesian model trained on some particular class. This suggests that by using this approach, we can infer those causally dependent features with their respective values which are high or even low probable for a class to appear. In BN approach, the extracted DSAP was a combination of two or more features conditioned on each other which come from causal semantic described in the model. Such correlation among features helped in narrowing down the search process for test cases satisfying presence of DSAP which thereby resulted in a good precision and recall. However in LDA, definition of low probable pattern was a single variable in some value. In other words, discovered set of low probable patterns by LDA were all independent of each other. Intuitively with this approach, probability of seeing test cases with the presence of any of these patterns would be large in number and therefore we obtained higher FP for LDA. More precisely, test cases satisfying conditional probability $P(X = x_i \mid Y = y_j)$ (Bayesian approach) will be much lesser than unconditional probability $P(X = x_j)$ or $P(Y = y_i)$ (LDA approach).

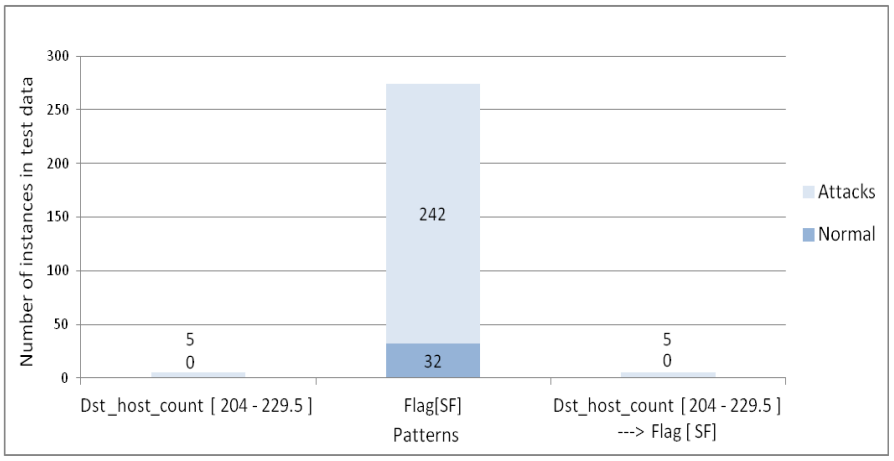
Below, we present several experiments in order to elaborate on why LDA approach resulted in poor precision in almost all data sets whereas, Bayesian approach performed reasonably well. To show this, we took few patterns resulted from LDA and Bayesian approaches on Lymphography data set. For deep insights, we preferred taking those patterns from these approaches in which features and their corresponding states were common. For example, two patterns mined using LDA technique on this data set was: (1) *Block_of_affere[yes]* and, (2) *Regeneration_of[yes]* respectively. Same features with

corresponding same values appeared in one of the DSAP extracted by the COM technique and the DSAP was : *Block_of_affere[yes]* $\rightarrow$ *Regeneration_of[yes]*. Though features and their respective values were same in the results, but, for LDA both features were considered independent while, second feature was conditioned on the first feature in the Bayesian approach. The bar graph in Figure 4.18a explains why the precision of LDA was poor but, better for BN on Lymphography data set. The X-axis of the graph shows the patterns extracted for this data set by both the techniques. The Y-axis shows the number of test cases satisfying presence of these patterns in individual classes, i.e., class for which BN and LDA were trained (addressed as "Normal") in the graph and the Anomaly class. For example, the number of instances that belongs to the "Normal" and "Anomaly" classes in the test set of this data set for the pattern: *Block_of_affere[yes]* were 11 and 5 respectively (the first bar). Similarly for the pattern: *Regeneration_of[yes]* distribution per class was 0 and 4 (the second bar). Based on two patterns mined by LDA on this data set, LDA will result in FP = 11. However for Bayesian approach it will be FP = 0.

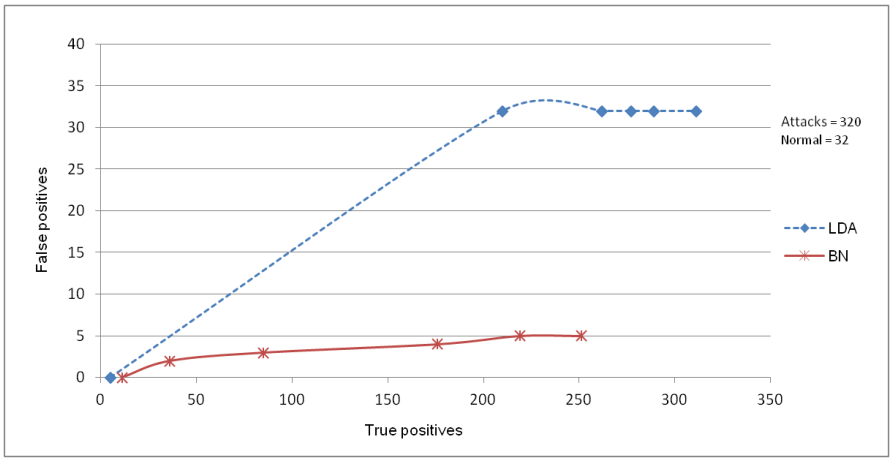
On similar lines as described for Figure 4.18a, in Figure 4.18b we present common patterns appearing in both COM and LDA approaches on KDD Cup* data set (a small randomized data set formed original KDD Cup data set discussed in Section 4.3.4.2). The graph in Fig. 4.18c shows the relationship between TP and FP with an increase in the number of patterns discovered from both BN and LDA techniques. The Top-6 low probable patterns were taken from the respective approaches and were tested on KDD Cup* . The graph shows the (TP, FP) achieved on each pattern when applied on test data for both LDA (indicated by blue dashed line) and BN (indicated by red line). Figure 4.18c shows that there was an increase in FP for LDA, which means that LDA marked all the normal instances as anomalies. On the other hand, BN performed reasonably well.



(a) Shows how relational dependency between variables helped reducing FP (third bar) as compared to when features were treated independently (first and second bar)



(b) Shows how relational dependency between variables helped reducing FP (third bar) as compared to when features were treated independently (first and second bar)



(c) Comparison of LDA and COM approach

Figure 4.18: Analysis and on COM and LDA approaches for outlier mining

4.4 Summary and Conclusion

In this chapter we proposed two robust probabilistic association rules which are based on causal knowledge captured by a Bayesian network (BN) to mine anomalous patterns for the domain. We extracted patterns which are examples of either *low support & high confidence* or *high support & low confidence* events. Extracted patterns were then tested on new data points to discover anomalies. We prove the credibility of our approach over existing well known outlier detection techniques by taking well known benchmark data sets and demonstrate that our approach is able to identify anomalies in high precision and recall. In addition, our approach can be used to discover contextual information from the mined anomalies, which other techniques often fail to do so.

Chapter 5

Mining Anomalies Using Hybrid Bayesian Networks

This chapter is based on following publication:

1. Mining Causal Outliers Using Gaussian Bayesian Networks
   Sakshi Babbar and Sanjay Chawla
   In Proceedings of the IEEE 24th International Conference on Tools with Artificial Intelligence,
   Athens, Greece, 2012, pp. 97-104

5.1 Introduction

Outliers are often characterized as entities which are "rare", "isolated," "distant from their nearest neighbors," or "low probability events." We first address to the questions: "how fair is to label every rare, isolated observation as an outlier?" and, "what are the signatures of outliers which makes them different from norm?". We start by giving a few examples which highlight the limitation of the standard orthodoxy in outlier detection.

Example 1: The following example is inspired from Griffith and Tennenbaum [35]. Suppose a fair coin is flipped four times. There are sixteen different equally likely possibilities. However the following two events: HHHH and TTTT, appear more surprising than the rest, even though each is equally like to appear as any other combination. Perhaps, implicitly, we were conditioned to expect an event with at least one head and one tail in the sequence. In which case the probability of observing such an event is $\frac{7}{8}$ vis-a-vis $\frac{1}{8}$ for a pure sequence. *Thus an event being classified as an outlier is often dependent on our implicit conditioning about the event rather than just low probabilities*.

Example 2: Now consider another synthetic example which highlights the possible relationship between peoples income and their expenditure for a certain region introduced in Chapter 1. *This small but intuitive example elaborate on why few, isolated observations may not be anomalies whereas, few observations lying near a dense cluster may be interesting anomalies*. In Figure 5.1, we present the two dimensional sample data explaining this scenario. For the discussion on this example, we assume variables income and expenditure are real valued. The X-axis in the figure represents income while, Y-axis expenditure. As observed, data points are roughly clustered. We name these clusters as C_1, C_2, C_3, C_4 and C_5. Dense cluster C_1 indicates that in a given region, high percentage of people spends in limit of what they earn. Unlike C_1, small probability of people spends as much as they earn, cluster C_2. Contrary to this, there are people who have high income but they choose to spend low, cluster C_3 indicates such situation. Finally, cluster C_4 and C_5 indicate situation where expenditure is higher than that of income. If objective is find outliers from sample data represented in Figure 5.1 using outlier detection techniques such as, distance [47] and density[17] based, then most probably these techniques may discover data points from clusters C_2

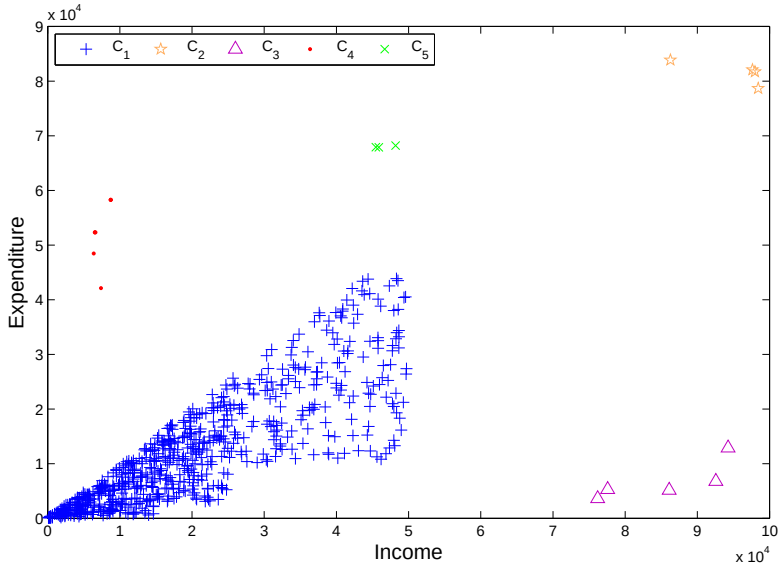


Figure 5.1: A synthetic example where X-axis represent income while Y-axis expenditure

and C_3 as potential outliers. Since data points forming these clusters are isolated and far away from their k nearest neighbors and so are easily detectable as candidate outliers. Discovering outliers from clusters C_2 and C_3 may be of least interest. Since these data points indicate common knowledge that people spending is bounded by their income. However, data points in clusters C_4 and C_5 are possibly more interesting as they represent situation where expenditure is higher income. *This small example motivates us to consider criteria other than for mining real outliers*.

The discussion above using examples suggests that outliers are not just unlikely events; perhaps they have some additional characteristics which make them different from the norm but also from the potentially false outliers. *An important characteristic which makes an observation a real outlier is its nature of violating common knowledge of the domain under investigation*. In Example 2, by common knowledge we know there is a relational dependency between income and expenditure. Usually it is uncommon to spend more than one earns. Therefore, labelling a high income and high expenditure event as an outlier is not interesting whereas a low income and high expenditure event is more interesting. Hence in order to discover true anomalies, causal semantic knowledge underlying the domain is important in the discovery process. Causal semantic knowledge refers to understanding the dependencies that exist between features of the

domain under examination.

In this chapter, we propose two techniques for mining interesting anomalies for application domains containing either set of all continuous variables or variables of different data types, i.e., combination of discrete and numerical variables on grounds of domain knowledge captured by a Gaussian Bayesian network (GBN for short) and *Hybrid Bayesian network* (HBN for short) respectively. Recall from Chapter 3, Gaussian Bayesian network is a kind of Bayesian network over continuous variables and whose parameters are defined using a probability density function. In contrast, Hybrid Bayesian network is a kind of BN which provide an ideal representation for capturing knowledge of the problem domain consisting of variables of different nature. The advantage of using Gaussian and Hybrid Bayesian networks over a standard Bayesian networks are two-fold. First, they provide capability of being useful in wider applications in comparison to a standard Bayesian network which may need discretization of continuous variables in order to use it. Second, by avoiding discretization, we could model domain knowledge using its true distribution of data.

In order to mine anomalies using Gaussian or Hybrid Bayesian framework, we exploited each independent factor induced by conditional independence property in BN defined by Equation 3.21 defined in Chapter 3 of this thesis. That is, we treated every factor, P(X | Pa(X)) captured by the Bayesian graph for revealing those data points which violate the *cause-effect* relation between parents and its child. In Figure 5.2 we present a revised version of Hybrid Bayesian network discussed in of Chapter 3 where three independent sets of *parent(s)-child* relationships are highlighted using a dotted, thick and a dashed arrows. We call these independent factors as *"causal subspaces"* (CS for short). To represent a causal subspace, we use the notation, ($X \rightarrow$ C) where the left hand side of arrow are the parents nodes of child node C presented on right hand side of arrow (similar concept of causal subspace was also introduced in Chapter 4). For example, three causal subspaces encoded in the model represented in Figure 5.2 are: (Employment $\rightarrow$ Lifestyle), (Employment, Income $\rightarrow$ Expenditure) and (Income, Expenditure $\rightarrow$ Mortgage). The reasons of exploiting each causal subspace for mining anomalies are multiple. First, every independent causal relationship can provide insights on meaning and degree of correlation among variables by which at a micro level, high and low probable events can easily be detected. Second, by studying each independent causal relationship, we can causally explain what makes an event normal/anomalous. Below we highlight two important characteristics of our definition of a

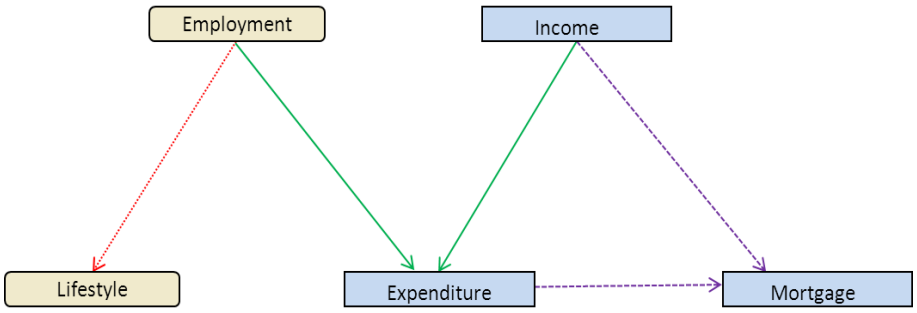


Figure 5.2: A synthetic Hybrid Bayesian network showing three causal subspaces highlighted by a dotted, thick and a dashed directed arrows

causal subspaces that will help understanding our methodology.

1. A child node in some causal subspace can be a parent node in another causal subspace. Similarly, a parent node in some causal subspace could be a child node in another causal subspace. For example in Figure 5.2, Expenditure is the child node in causal subspace (Employment, Income $\rightarrow$ Expenditure) while it becomes parent node in causal subspace (Income, Expenditure $\rightarrow$ Mortgage).

2. In any causal subspace there could exist more than one parent node while, there can be only one child node.

For now we assume that we are given graphical layout of Bayesian network and set of parameters associated with each node. Based on this framework, we propose to apply *causal reasoning* in each CS for every test case in order to check if it violates the encoded causal semantic knowledge. And if causal knowledge is violated then, data point is considered outlier where concerned causal subspace explains the reason of outlierness. We explain using a small example how causal reasoning can help evaluating a data point, refer BN in Figure 5.3. Two nodes represent Income and Expenditure, and direction of arrow indicates that expenditure is causally dependent on income. Here we assume that parameters, mean and standard deviation associated with these nodes capture trend of relationship of data samples clustered in C_1, C_2 and C_3 of Figure 5.3. Given this information, graph in Figure 5.4 shows pattern of mean in expenditure given

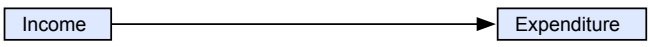


Figure 5.3: Bayesian network on a income-expenditure example

income using causal inference in Bayesian network shown in Fig. 5.3. The X-axis

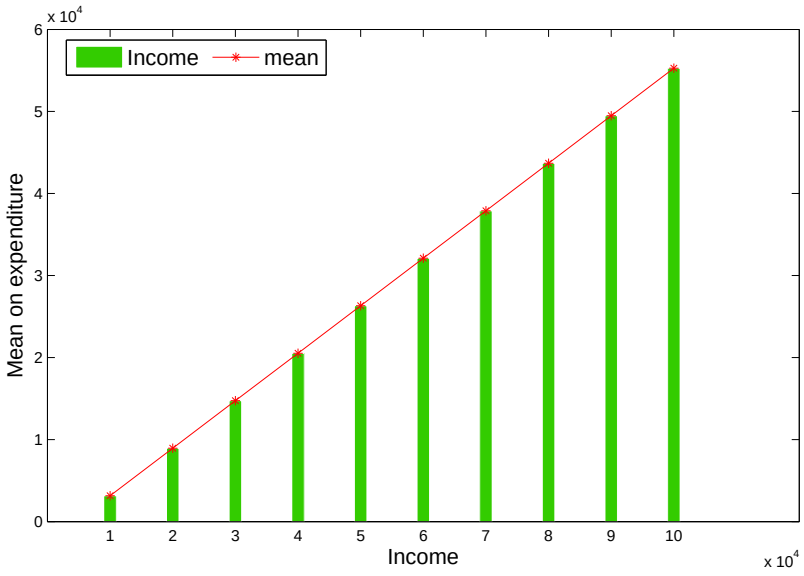


Figure 5.4: Shows pattern of mean in expenditure (Y-axis) given income (X-axis) using causal inference in Bayesian network

of the graph represents income in thousands whereas; change in mean of expenditure given income is reflected by the Y-axis. A trend line clearly shows how mean at expenditure grows with increase in income. However, more interesting is by making use of using causal inference we can clearly infer expected mean on expenditure, given income through this graph. For example if income is around 80K, mean on expenditure is expected to center around 45K. In order to decide if a test case t = {Income = i, Expenditure = e} is anomaly, we propose to apply causal inference by entering known value in income (i.e., i in the testcase t) and measure how far given value of expenditure (i.e., e in the testcase) is from the change in mean at expenditure. Higher the deviation, higher the chances of a test case to be an anomaly.

5.1.1 Problem Statement

The problems that we address in this chapter are as follows:

1. What makes an outlier really interesting? What are the key characteristics of an interesting anomaly which differentiate it from the false outlier?

2. Given a data set, find those data points which under domain knowledge are low probable to occur.

3. From set of discovered data points as anomalies, explain why they are unusual.

To solve the above problems, we propose integration of domain knowledge in the discovering process. We show that outliers that those events which violate common knowledge of the domain. Using an appropriate Bayesian network to capture domain knowledge, we present an algorithm which not only identify outliers, but also explain their anomalous nature.

5.1.2 Contributions

In this chapter we make following contributions:

1. We present a measure to discover outliers in numerical data sets and data sets containing mixture of data types on the grounds of domain knowledge captured by a Bayesian framework. Our technique not only identifies real outliers, but also presents subset of attributes that explains what makes an event suspicious. Such explanation contributes to a vital knowledge for the domain by which domain can learn and establish more improved system against anomalies.

2. By exploiting parametric information encoded with every node in the Bayesian framework and causal inference, we propose approaches which can mine anomalies from given data sets with high accuracy.

5.1.3 Notations and Basic Concepts

In this chapter all notations corresponding to Bayesian networks are followed from Chapter 3. However, notations specific to this chapter are summarized in Table 5.1.

Notation	Description
d	A data point
X_τ	Set of discrete nodes in BN
$X_\diamond$	Set of continuous nodes in BN
CS	Causal subspaces in BN
$\|CS\|$	Total number of causal subspaces in BN
$CS_\diamond$	Causal subspaces involving only discrete nodes
CS_τ	Causal subspaces involving discrete and continuous nodes

Table 5.1: Notations and basic concepts

The chapter is organised as follows. In Section 5.2, we present our detailed methodology. Our experiments and analysis on results are explained in Section 5.3. Finally, in Section 5.4 we conclude the chapter. Table 5.1 lists all notations used in this chapter.

5.2 Anomaly Detection Using Gaussian & Hybrid Bayesian Networks

Recall from Chapter 3 that in Gaussian Bayesian framework, each independent variable is represented by a pair (μ, σ), where μ and σ represents variables mean and standard deviation. Whereas, mean of a dependent variable depends linearly on its parents and is represented by the Equation 3.27 defined in Chapter 3. On Bayesian inference, each node is parameterized with a new information in the form a pair (μ', σ') where, μ' and σ' represents mean and standard deviation indicating how nodes are influenced by each other in the network. Likewise in Hybrid Bayesian networks, after inference each node is represented by a mean and standard deviation. However, the methodology of computing mean and standard deviation on a node is computed in different ways for two different setting of Bayesian framework (refer Chapter 3).

We assume throughout this section that Bayesian graphical structure[1] and associated parameters are given. In practice, to find if a data point d is anomalous, we adopted the following strategy: in every causal subspace, we apply causal inference by entering known observations in parent nodes which in result reveals expected mean and variance at the child node.

In other words, if we wish to know whether the causal relationship, (X = $x_i \rightarrow$ C

[1] we refer Gaussian and Hybrid Bayesian models as Bayesian networks unless not stated explicitly in this chapter

= c_j) holds for the data point d then, c_j should center around μ_C (mean at C) after entering the evidence X = x_i. In order to compute deviation between expectation and mean computed after inference, we propose to use Z-score test. A Z-score is a statistical measure to compute how far a data point is from the mean. Higher the Z-score, more likely are the chances for a data point to be generated by a different mechanism and hence, high probability of being an outlier. Equation 5.1 explains how Z-score for a child node C at every causal subspace i is computed.

$$Zscore(C_{CS_i}) = (C^d_{CS_i} - \mu_{C_{CS_i}})/\sigma_{C_{CS_i}} \tag{5.1}$$

Where notation :
$C^d_{CS_i}$: refers to value of child node C of i^{th} causal subspace in data point d
$\mu_{C_{CS_i}}$: mean at child node C of i^{th} causal subspace after inference
$\sigma_{C_{CS_i}}$: standard deviation at child node C of i^{th} causal subspace

Since our aim is to discover causal outliers, so we investigated every i^{th} causal subspace and entered as evidence known values in the parent nodes which on principle of causal inference resulted in an expected mean at child node, represented by $\mu_{C_{cs_i}}$ in Equation 5.1. Based on causal semantics of i^{th} causal subspace, value of child node in the data point represented by $C^d_{CS_i}$ in Equation 5.1 should center around mean computed after inference which is $\mu_{C_{CS_i}}$.

We now explain our methodology on income-expenditure example. Using Figure 5.5 we show how causal inference in BN represented by Figure 5.3 together with Equation 5.1 be used in order to discover those data points which violate causal semantic. For this purpose, we created two sample data sets for this example, i.e., sample data set 1 and sample data set 2. Value of income in both these data sets is same while, few abnormal values for feature expenditure were created against income in the sample data 2. The X-axis of the graph in Figure 5.5 represents mean on expenditure calculated after entering known value in income given by the sample data set using causal inference. On the other hand, Y-axis indicates Z-score computed using Equation 5.1 on values of expenditure present in the given sample data sets. Pattern of Z-scores achieved on two sample data sets are shown by a dashed and a thick line.

Clearly for sample data set 1 (thick line), Z-score computed on expenditure is very low for every test case belonging to this data set. Whereas, for sample data set 2,

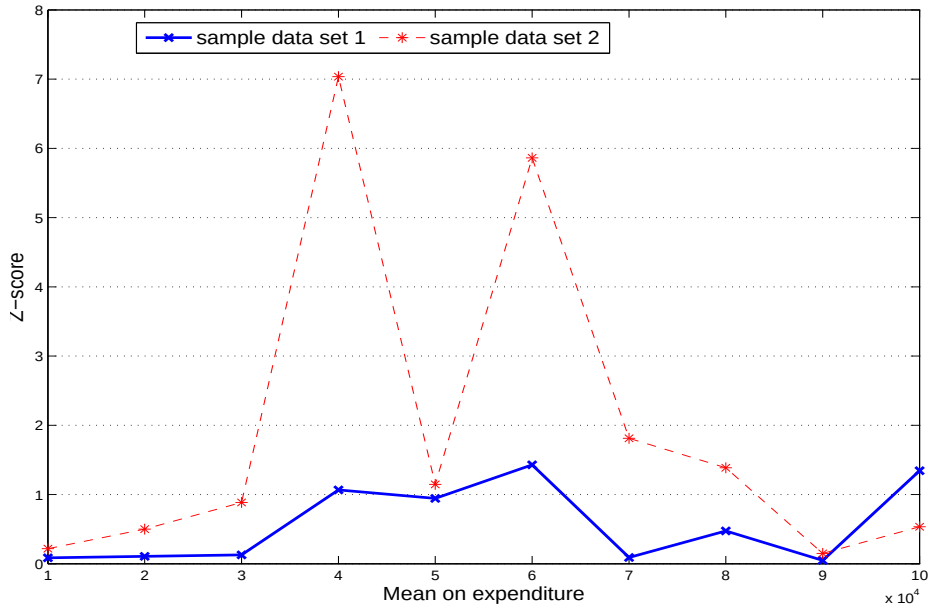


Figure 5.5: In comparison to sample data set 1, sample data set 2 carries two anomalies indicated by 4^{th} and 6^{th} data points.

indicated by a dashed line, Z-score computed for two test points namely, fourth and sixth is quite high which shows these data points do not capture intended pattern of relationship between income and expenditure. Perhaps these data points violate causal relationship and hence are outliers. Since the actual value corresponding to a Z-score signifies how far a data point is from mean regardless of whether value is below or above mean, i.e., negative or positive so, we used absolute value of Z-score in Equation 5.1. After computing Z-score at every i^{th} causal subspace, we added all Z-scores to form a score for the data point d, refer Equation 5.2. Where $|\mathrm{CS}|$ is the total number of causal subspaces present in the BN. And, later these scores are sorted. Top n high scored test cases are treated as potential causal outliers. To investigate on domain specific anomalous causal subspaces for discovered top n outliers, we simply study their Z-score on every causal subspace. A substantially higher Z-score explains the source of anomaly.

$$score(d) = \sum_{i=1}^{|CS|} Zscore(C_{CS_i}) \tag{5.2}$$

This simple example is a case of pure Gaussian setting. However, in Hybrid Bayesian networks, there can exist any number of discrete and continuous variables connected in

an arbitrary way. Such an environment results in a complex setting where variables in each causal subspaces needs to be analysed for their data types before applying the process of anomaly detection. Consider a HBN and let it consist of set of variables $X = \{X_1, X_2, ..., X_n\}$ be partition into set of discrete node (represented by $X_\diamond$) and, the set of continuous nodes (represented by X_τ). Then for each $X_i \in X_\diamond$ in the model there exist a conditional probability distribution whereas, each $X_j \in X_\tau$ is defined using probability density function. Like in the case of Gaussian BNs, we exploited parametric information associated with each CS in the HBN model to reveal data points which violates causal relationship using causal inference. Interestingly, in HBN there could exist following three types of relationships among variables depending upon their data types.

1. ***A pure discrete case:*** An environment where discrete parent nodes are connected to a discrete child node. We use notation $X_\diamond$ to represent set of discrete nodes involved in a pure discrete causal subspaces. Total number of pure discrete causal subspaces is denoted by $|CS_\diamond|$. An example of $CS_\diamond$ in Figure 5.2 is, (Employment $\rightarrow$ Lifestyle).

2. ***A mixture of discrete and continuous case:*** A case where number of discrete and continuous parents are linked to a continuous child node. Such causal subspaces are defined using notation CS_τ, and $|CS_\tau|$ is used to represent total number of causal spaces involving mixture of variables. Causal subspace, (Employment, Income $\rightarrow$ Expenditure) in Figure 5.2 contains a one discrete variables while, there are two continuous variables.

3. ***A pure continuous case:*** Where a child node is conditioned on one or more continuous parents. We use common notation of CS_τ to represent causal spaces involving continuous case. One example of this case is, (Income, Expenditure $\rightarrow$ Mortgage).

For a causal subspace containing all discrete variables, we used methodology based on two probabilistic rules ($\mathbf{R}_1$ and $\mathbf{R}_2$) presented in Chapter 4. However, there exist few limitations in the parameter *minsupp* (minimum support) used in definition of $\mathbf{R}_1$. Below we present several improvements over definition of *minsupp* parameter.

1. There exist a limitation in situation when for all $x_i \in Val(X)$, the $P(X = x_i)$ is equal. That is, the case where for all $x_i \in Val(X)$, $P(X = x_i) = \frac{1}{|Val(X)|}$. It would

be unfair to treat *maxsupp* with the same probability and as *minsupp*. We propose that in such cases *minsupp* parameter should be treated as null.

2. Based on definition of *minsupp* parameter, there can always be only one such state x_i for which *minsupp*(X = x_i) exist. However, in case of high entropy, there could exist more than one state with equally low probability of occurrence. In order to consider more than one low probable state, we propose use of a common low support threshold for all the parent nodes in the network. For example, *minsupp* set to 10% will consider all states of a variable occurring with less than 10% probability.

Based on the definition of rules $\mathbf{R}_1$ and $\mathbf{R}_2$ described in Chapter 4 of this thesis and modifications made in *minsupp* parameter defined above, we exploited each discrete causal subspace using rules. Let in any $CS_i \in CS_\diamond$ there exist *r* anomalous patterns, then score of *r* patterns in CS_i is formed using Equation 5.3. The score ($Score_\diamond$ in equation) is computed by multiplying conditional probability (confidence) with the prior probability (support). Reason of using logarithm is discussed shortly. After computing score at each causal subspace, all scores are added to form a combine score for all discrete causal subspaces encoded in the model for a test point d, refer Equation 5.4.

$$Score_\diamond(CS_i)_{(i \in CS_\diamond)} = \log \sum_{k=1}^{r} P(C|Pa(C))P(Pa(C)) \tag{5.3}$$

where C and Pa(C) $\in X_\diamond$

$$Score_\diamond(d_{CS_\diamond}) = \sum_{l=1}^{|CS|_\diamond} Score_\diamond(CS_l) \tag{5.4}$$

For causal subspace with mixture of data types, we resort to concept of Z-sore. However, Z-score at every child node C $\in X_\tau$ in each CS_τ of HBN is computed using Equation 5.5 instead of Equation 5.1.

$$Zscore_\tau(CS_j)_{(j \in CS_\tau)} = \begin{cases} \log(1 - \frac{C^d_{CS_{\tau_j}} - \mu_{C_{CS_{\tau_j}}}}{\sigma_{C_{CS_{\tau_j}}}}) & \text{if } \frac{C^d_{CS_{\tau_j}} - \mu_{C_{CS_{\tau_j}}}}{\sigma_{C_{CS_{\tau_j}}}} < 0 \\ \log \frac{C^d_{CS_{\tau_j}} - \mu_{C_{CS_{\tau_j}}}}{\sigma_{C_{CS_{\tau_j}}}} & \text{if} \frac{C^d_{CS_{\tau_j}} - \mu_{C_{CS_{\tau_j}}}}{\sigma_{C_{CS_{\tau_j}}}} > 0 \end{cases} \tag{5.5}$$

$$Score_{\tau}(d_{CS_{\tau}}) = \sum_{o=1}^{|CS_{\tau}|} Zscore_{\tau}(CS_o) \tag{5.6}$$

After computing Zscore_{τ} at every continuous child node in CS_{τ} appearing in test point d, all scores are added to form a combine score, refer Equation 5.6. The overall score of a test point d is computed by adding scores from Equation 5.4 and Equation 5.6 as represented by Equation 5.7. The logarithm function is used in Equations 5.3 and 5.5 so as to bring the combined score in one scale. In a discrete framework, Equation 5.4 will yield result in the range of [0-1] if logarithm function is not used. Similarly, scale of Z-score (Equation 5.5) will range between [-3σ, +3σ] in the absence of logarithm function. Hence to normalize the score, we used logarithmic function. For a given test data, scores of each test point is computed using Equation 5.7. Later, scores obtained are sorted. Top n high scored test cases are treated as anomalies. To investigate on explanation of discovered top n anomalies, we simply study their score computed by Equations 5.3 and 5.5. A positive score computed in Equations 5.3 explains presence of anomalous pattern existing among discrete variables of the test case whereas, a substantially higher Z-score computed using Equation 5.5 describe the source of anomaly in a continuous framework.

$$Score(d) = Score_{\diamond}(d_{CS_{\diamond}}) + Score_{\tau}(d_{CS_{\tau}}) \tag{5.7}$$

5.2.1 Algorithm

Algorithm 2 describes procedure of mining anomalies in Gaussian and Hybrid Bayesian setting. We call our algorithm COMGN which stands for causal outlier mining in Gaussian networks.

The computational complexity of our algorithm COMGN is governed by two main factors: (1) Size of the test data and, (2) Inference in GBN or HBN. The problem of inference in BN is NP-hard [48], and therefore it probably requires exponential time in the worst case. However, in practice strategies such as variable elimination method implemented via message passing technique can tackle real-world applications very effectively. In particular, for efficient inference process, intermediate factors in a Gaussian networks can be described compactly using a simple parametric representation called the canonical form which is closed under basic inference operations allowing inference

process to be define using simple data structures [48]. Intuitively, in our approach, we are not performing very complex inference of the form P(C | Z) where, Z represents set of variables other than parent variables of C which may require complex marginalization operations. Instead, we are using very simple inference of the form P(C | Pa(C)) where, Pa(C) define parents of child node C.

Algorithm 2 COMGN

Input: BN, Bayesian network (GBN or HBN);
minsupp, minimum support threshold;
minconf, minimum confidence;
maxconf, maximum confidence;
n, number of outliers;
D, test data;

1: **for** $(each\ d\ in\ D)$ **do**
2: **while** $(i < |CS|)$ **do**
3: **if** $(CS_i \in CS_\diamond)$ **then**
4: $Compute\ Score_\diamond(CS_i)\ using$ Equation 5.4
5: **else**
6: **if** $(CS_i \in CS_\tau)$ **then**
7: $Compute\ Score_\tau(CS_i)\ using$ Equation 5.6
8: **end if**
9: **end if**
10: $i = i + 1$
11: **end while**
12: $Compute\ Score(d)\ using$ Equation 5.7
13: **end for**
14: $Output\ top\ n\ outliers$

5.3 Experiments, Results and Discussion

In this section we report on experiments that we carried out in order to mine anomalies present in the data set using Gaussian and Hybrid Bayesian framework.

5.3.1 Experimental Setup and Data sets

The prerequisite for our approach is the need of Bayesian models which can reveal causation that exist in a domain. Recall from Chapter 3 that Bayesian modeling can be

achieved by either consulting domain experts or using learning algorithms. In our work, we used learning algorithms in order to learn Bayesian networks. We used two structure learning packages in R namely Deal [32] and Bnlearn [62]. Deal package can learn structure from data set containing mixture of data types whereas; Bnlearn offers several algorithms for learning structure from a pure numerical data set. We also used Bayes Net Toolbox [42] in Matlab for the purpose of coding. For every data set we used for experiments, training and testing framework was designed where, the training set was used for learning Bayesian models, and test set for evaluation. The following protocol was adopted for designing such setting for each data set used in our experiments five times. The process is also summarized in Figure 5.6.

1. Given data set was divided into 80% training set and 20% test set.
2. Bayesian model was learnt using training set.
3. Test data was pre-processed to create anomalies artificially by taking following steps:
    - The test set was randomly partitioned into two sets: *setCreateAnomalies*, which contained 5% of records from the test set and rest of the records where grouped in a set called *setNormal*.
    - Further, for each data point in *setCreateAnomalies*, we randomly chose *m* child nodes. For a every discrete child node selected, we replaced its current state with its randomly chosen state. However, for a continuous child node, we added to it, its mean computed from the test set. The converted data points were grouped under a set called *setAnomalies*.
    - The test data was recreated by the union of sets: *setAnomalies* and *setNormal*.

5.3.1.1 Data sets

Table 5.2 summarizes list of data sets used in our experiments. The Column 1 of the table presents data set names. To distinguish between a numerical data sets and data sets containing mixture of data types (discrete and continuous variables), we lists real valued data set names using a simple text whereas, names of other set of data sets are

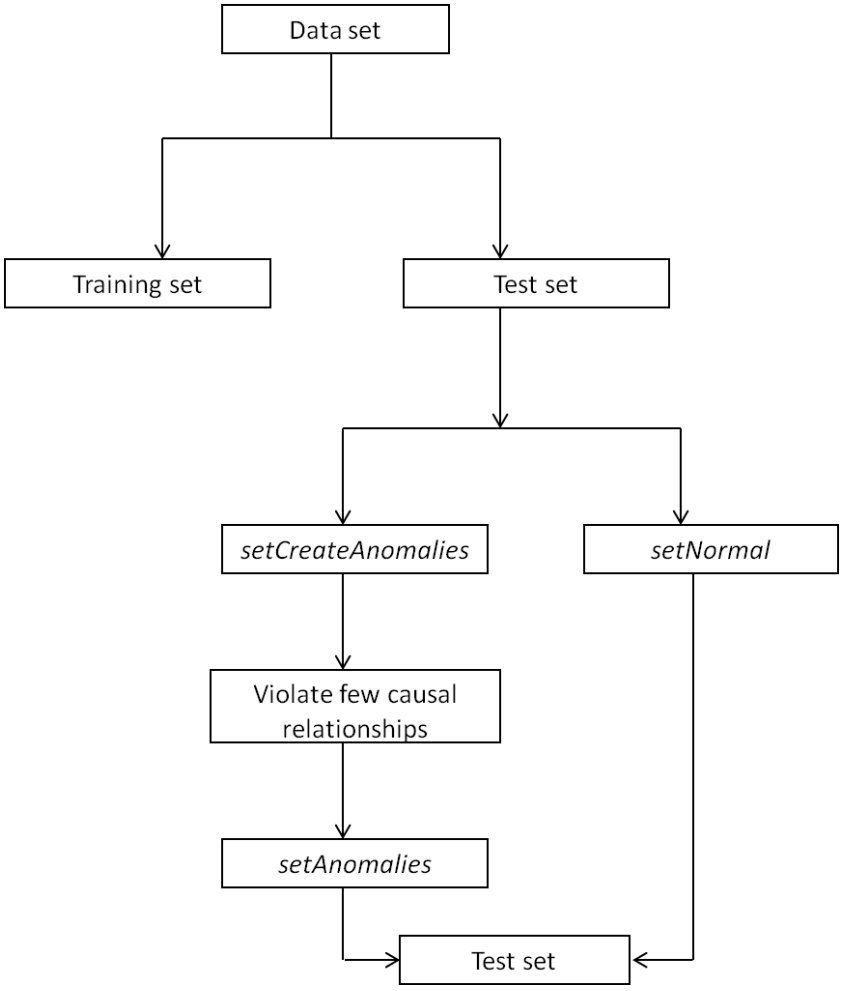


Figure 5.6: Experimental setup

Data set	**(R x D)**
Body fat	(253 x 15)
Ecoli	(153 x 7)
Breast cancer	(356 x 32)
Sonar	(208 x 60)
Slump	(103 x 10)
Musk	(6598 x 169)
Spectometer	(531 x 103)
Boston	(506 x 13)
Waveform	(5000 x 40)
Heart-C	(274 x 12)
Hypothyroid	(3774 x 28)
NHL basket ball	(548 x 8)
KDD Cup	(141480 x 41)

Table 5.2: Description of data sets

represented using an italic notation. The notation (**R x D**) in Column 2 of the table states that there were **R** number of instances and **D** number of features present in the data set. All data sets except NHL basket ball, Body fat and Boston were taken from UCI repository [4]. NHL basket ball was taken from [1]. And data sets, Body fat and Boston from CMU Statlib repository [2].

For KDD Cup data set in particular, we did not create anomalies artificially because for this data set we knew if particular instance is anomaly. We trained Bayesian model on 80% normal instances and rest 20% normal records were integrated with attacks to form a test set. This data set contains 22 different attack types. In Appendix A we present list of these attack.

5.3.2 Results

We first present Bayesian networks learnt over few data sets. Then, results achieved on various data sets are reported. We also present few subspaces discovered by algorithm COMGN on KDD Cup data set which explains why the discovered data point is anomaly. Thereafter, we present a detailed examination of 22 attack types present in KDD Cup data set to show how our technique responded on individual attack type.

5.3.2.1 Bayesian networks learnt

In Figures 5.7, 5.8, 5.9 and 5.10 we show Bayesian networks learnt over four data sets namely, Ecoli, Boston, NHL basket ball and KDD Cup. Names given to nodes encoded in all Bayesian networks are same as defined in respective data sets. In Appendix A, description on these variables are given. In order to distinguish discrete variables from continuous variables in the model, we followed the convention of representing a discrete variable using a rounded box while, continuous variable using a square box. For few data sets, for example, KDD Cup out of total 41 attributes (refer Table 5.2) 33 attributes were found connected in its Bayesian structure. This implies, there was no relation among remaining 8 attributes with those 33 attributes modeled in the network.

5.3.2.2 Experimental evaluation

We compared performance of our technique with following two classical outlier detection techniques.

1. k^{th}-NN (k^{th} nearest neighbour) [47] outlier detection technique with parameter k set to 5. In order to make it applicable for a training/testing setting, we changed the method sightly. Instead of scorning a test point on the basis of its deviation from all other data points belonging to the test data, we scored a data point to extent of which it deviates from the training set. Given a training data set, in order to use the 5^{th} to determine the degree to which a test point is anomalous, we simply use the distance from the point to its 5^{th} in the training set. *A larger distance indicates a more anomalous point*

2. Local outlier Factor (LOF) [17] anomaly detection. LOF was also modified in the similar way as k^{th}-NN. The LOF of each test point is computed with respect to the training data set in order to score the point.

Table 5.3 summarizes result we achieved on thirteen data sets using COMGN, k^{th}-NN and LOF techniques. Column 1 lists name of the data sets. In column 2 of the table we present, average precision/recall achieved by COMGN. Following columns list results obtained using k^{th}-NN and LOF approaches over same data sets respectively. As indicated by results, LOF did not perform well on almost all data sets. The k^{th}-NN results especially for Ecoli and Breast cancer data sets were not encouraging. In comparison to these techniques our approach worked reasonably well on almost all

data sets giving accuracy of more than 75%. We further investigated on why k^{th}-NN and LOF performed so poorly on few data sets in comparison to COMGN which we will discuss in the next section.

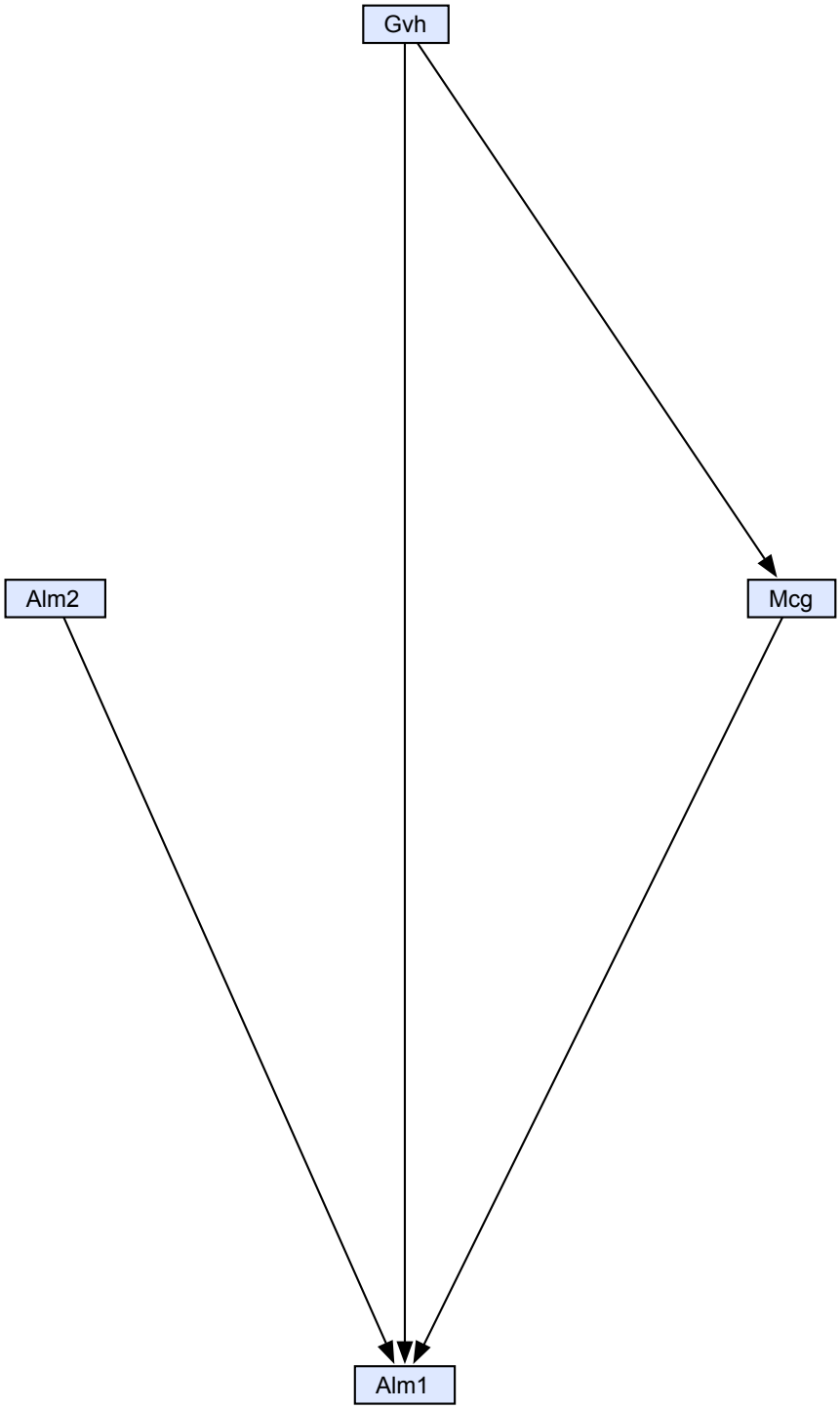


Figure 5.7: Bayesian network on Ecoli data set

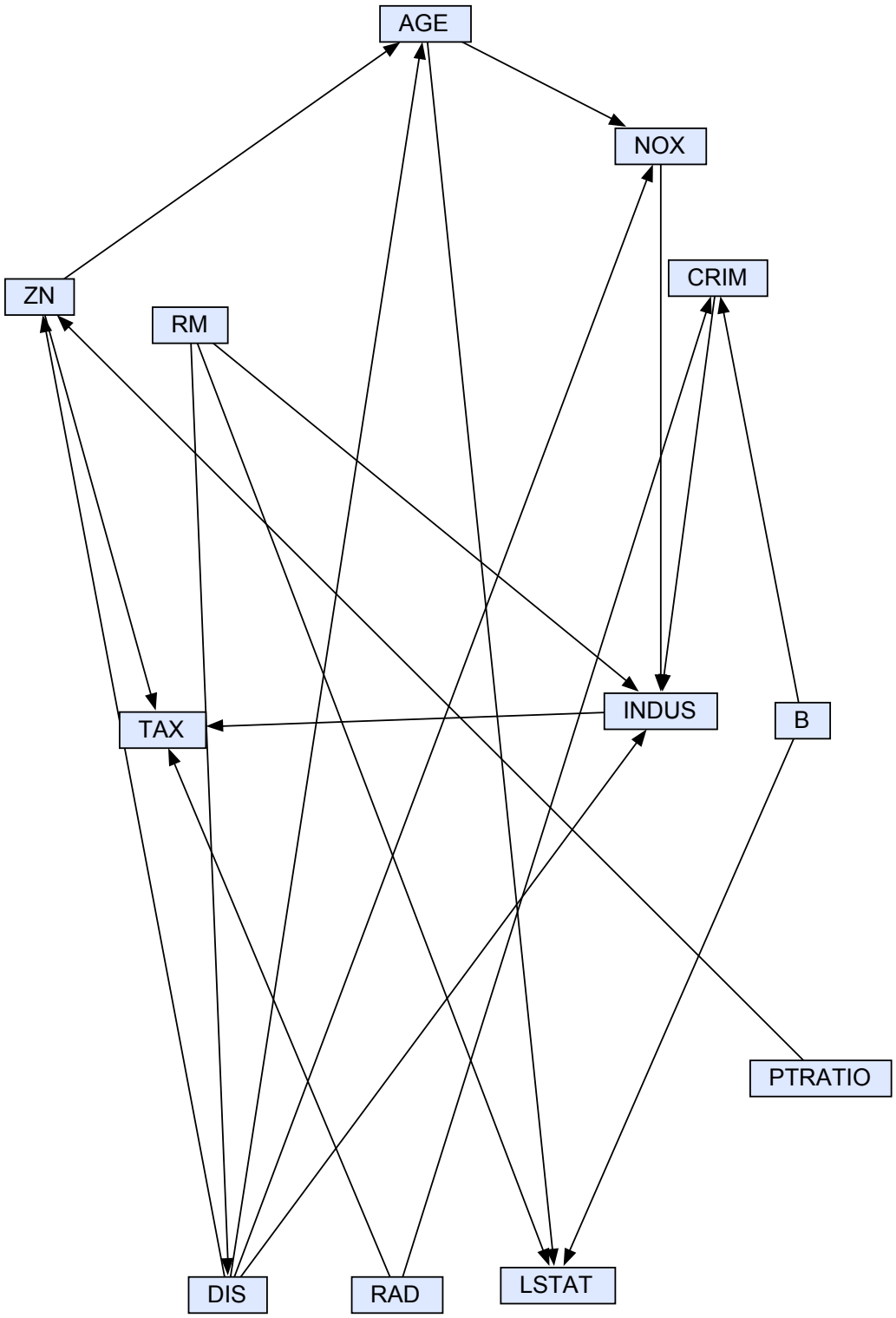


Figure 5.8: Bayesian network on Boston data set

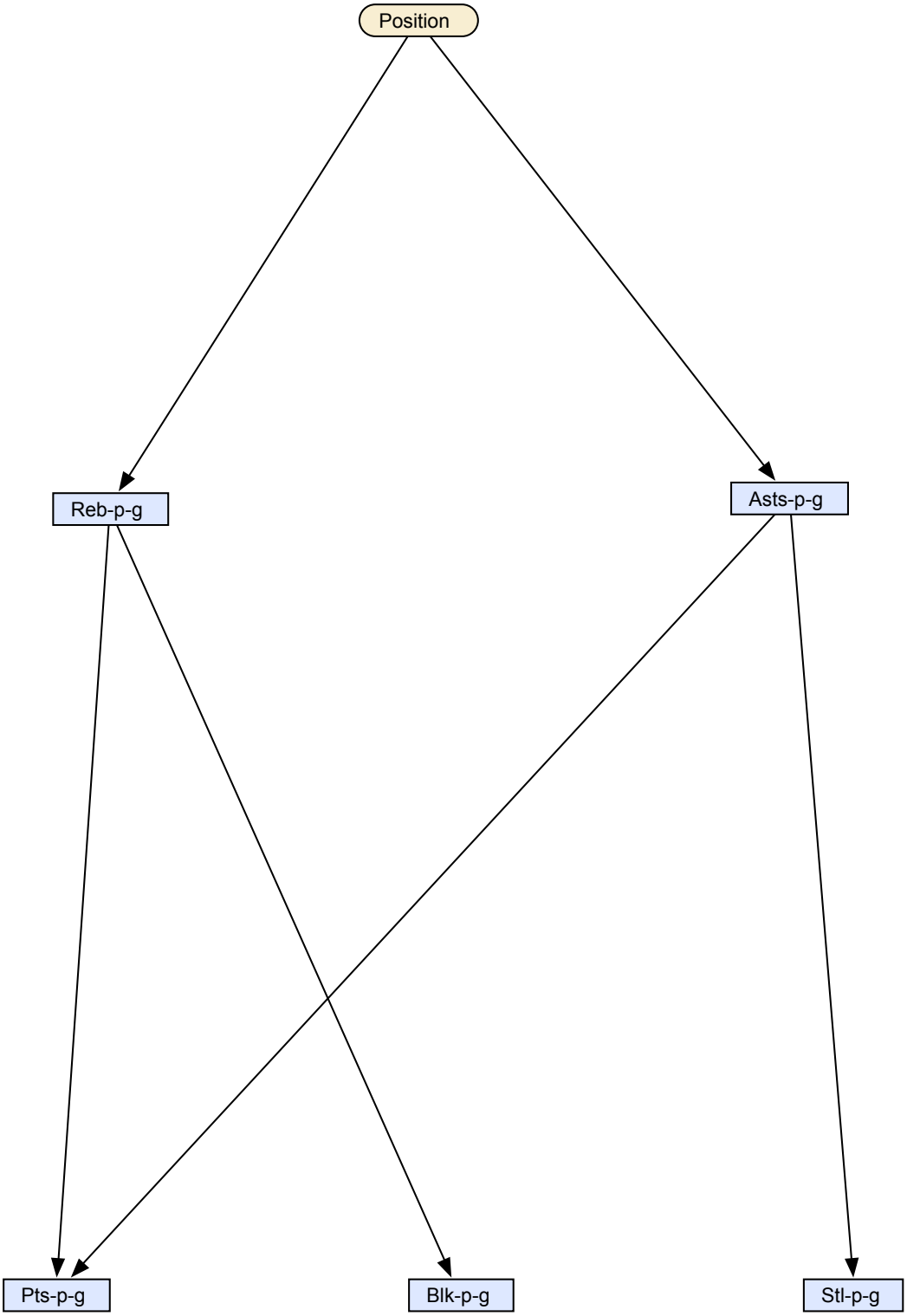


Figure 5.9: Bayesian network on NHL basket ball data set

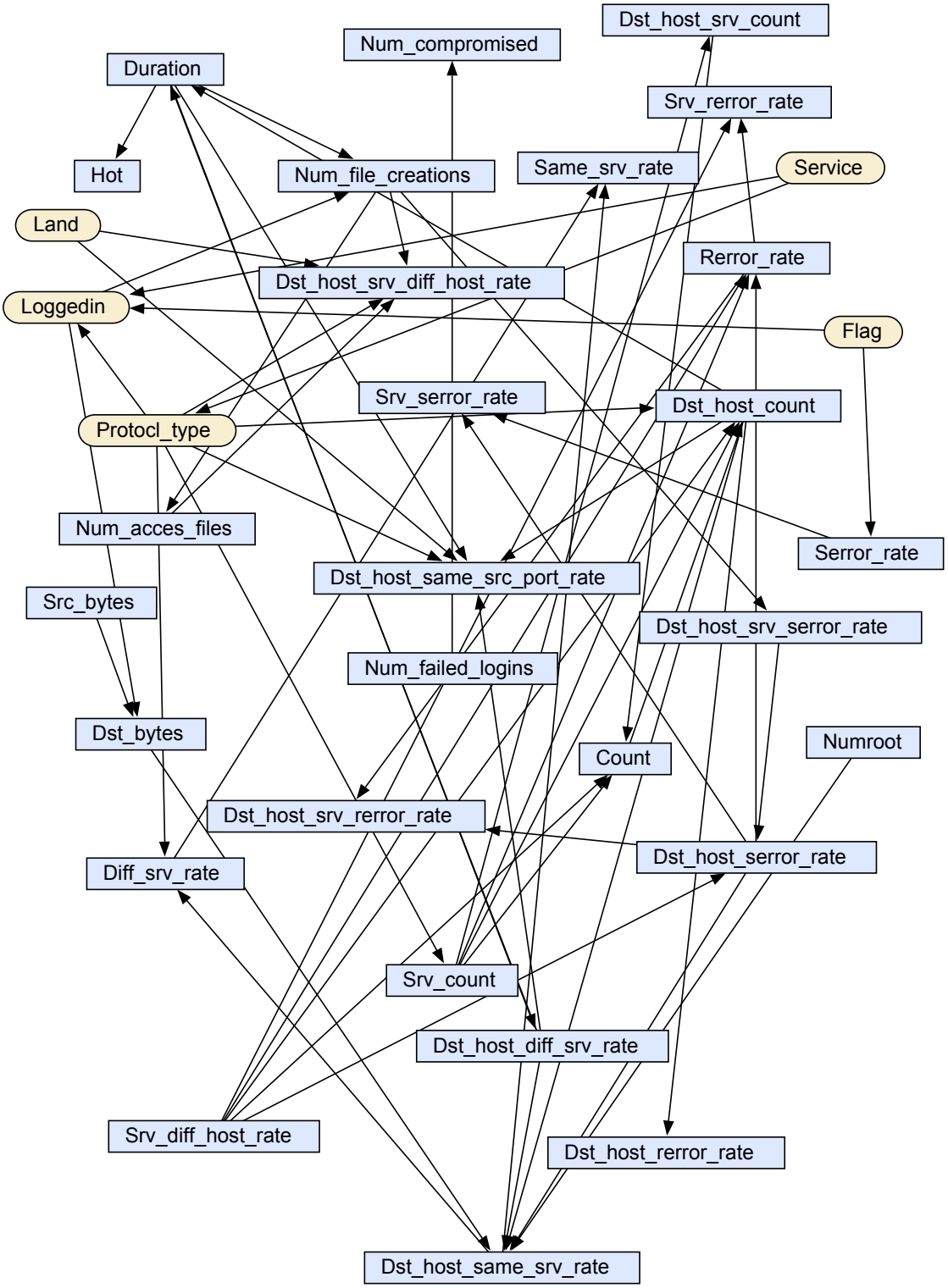


Figure 5.10: Bayesian network on KDD Cup data set

Data sets	Avg. precision/recall		
	COMGN	k^{th}-NN	LOF
Body fat	.76	.72	.56
Ecoli	.93	.38	.38
Breast cancer	.79	.47	0
Sonar	**.90**	1	.82
Slump	**.86**	.86	.72
Musk	.77	.66	.61
Spectometer	.82	.60	.61
Boston	.67	.82	.56
Waveform	.77	.71	.64
Heart-C	.82	.69	.52
Hypothyroid	**.87**	.61	.31
NHL basket ball	**.86**	.73	.51
KDD Cup	**.98**	.71	.68

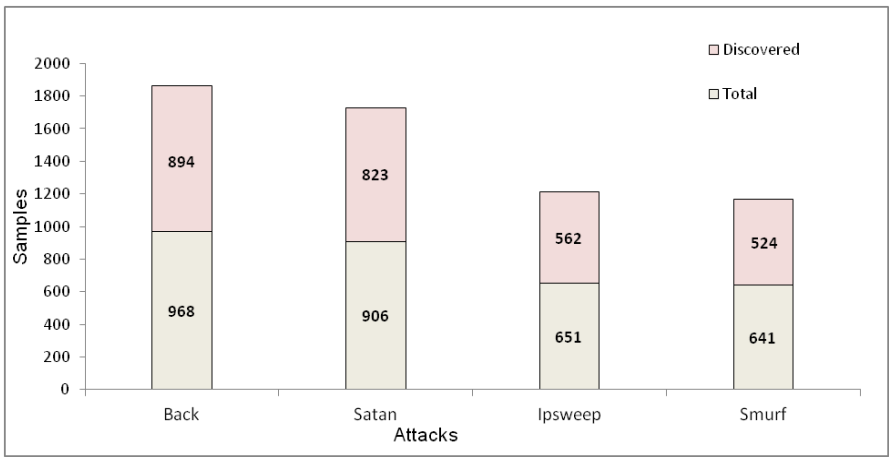
Table 5.3: Result on different data sets achieved using COMGN, k^{th}-NN and LOF algorithms

5.3.2.3 Analysis on KDD Cup intrusion detection data set

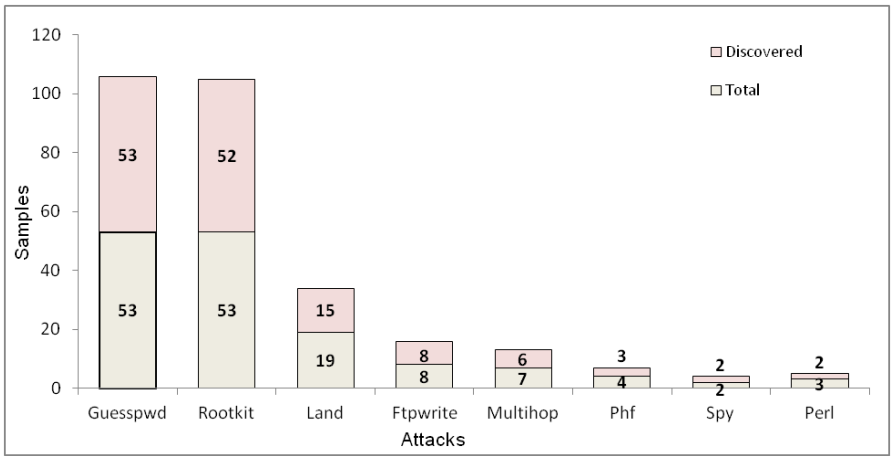
We studied in detail results we achieved on KDD Cup data set to know how our technique performed on 22 different attack types. The list of 22 attack types and their quantity in data set is presented in the Appendix A for reference. The configuration of 22 attacks contained in this data set is very imbalanced. For example, 82% of attack samples belong to attack type *Neptune* whereas, there are few attacks such as *Spy*, *Perl*, *Phf* which has less than 5 samples. So, if an anomaly detection technique gives recall of more than 75% then it would be hard to say about robustness of the approach since detection of Neptune alone can give high recall. In our work, we studied each attack type individually to see how our technique performed in detecting them, and to mine reasons of their anomalous nature.

In Figures 5.11a, 5.11b and 5.12a we present, 13 attacks types for which our technique succeeded. The X-axis represent attacks while, Y-axis are their number of samples present in the test set and those discovered by our technique. Each lower bar on the top of each attack is the total number of samples present in the data set whereas, upper bar is the discovered number of samples using our approach. In addition to this, we present insights on false positive, i.e., number of normal instances discovered as attacks by our technique in Figure 5.12a. Out of 22 attacks, COMGN succeeded on 13 attacks. There were few attack types for example, *Teardrop*, *Warezclient*, *Nmap* and

Buffer overflow for which recall of our approach was less than 50%, refer Figure 5.12b. However, for five attacks namely *Portsweep*, *Warezmaster*, *Pod*, *Loadmodule* and *Imap* our technique failed to detect them.

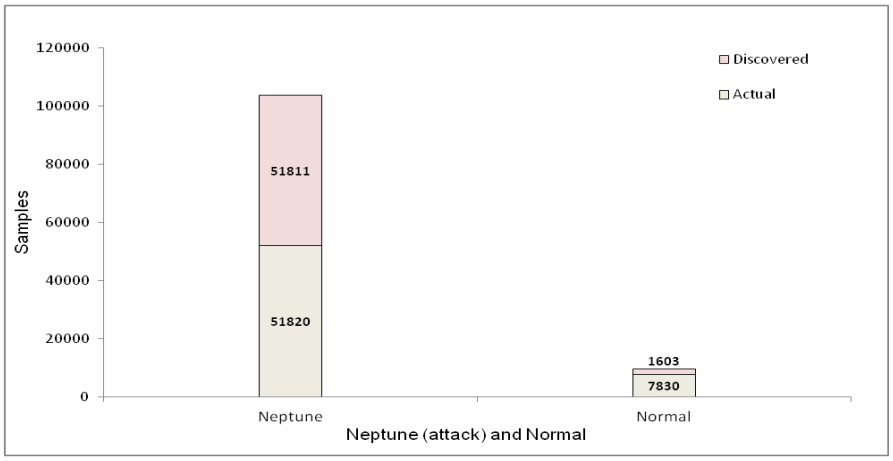


(a) Number of samples of attacks: *Back*, *Satan*, *Ipsweep* and *Smurf* discovered by COMGN

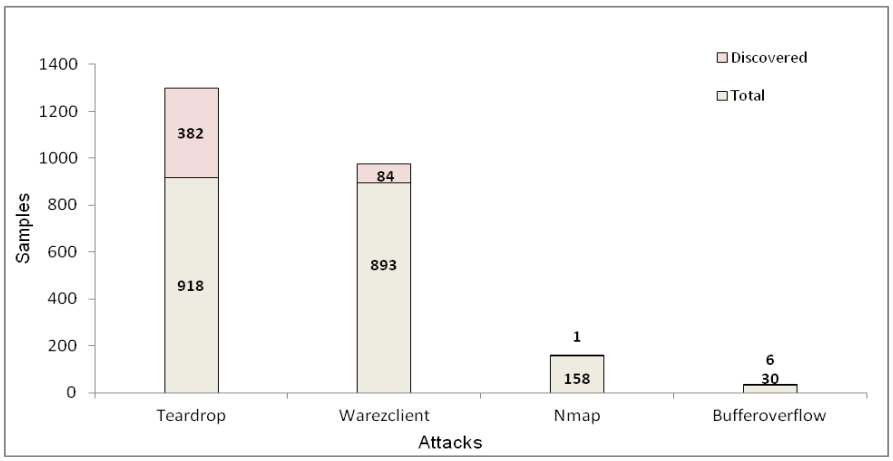


(b) Number of samples of attacks: *Guesspwd*, *Rootkit*, *Land*, *Ftpwrite*, *Multihop*, *Phf*, *Spy*, *Perl* discovered by COMGN

Figure 5.11: Performance of COMGN on DOS, U2R, R2L and Probe attack types



(a) Number of samples of attack: *Neptune* and normal instances discovered by COMGN



(b) Number of samples of attacks *Teardrop*, *Warezclient*, *Nmap* and *Bufferflow* discovered by COMGN

Figure 5.12: Performance of COMGN on DOS, U2R, R2L and Probe attack types and false positives

5.3.2.4 Anomalous patterns discovered

In Table 5.4 we present two anomalous causal subspaces each for few attack types discovered by our technique. The discovered subspaces explain set of attributes which are targeted by attacks. For example Table 5.4 reveals the fact that *Smurf* (attack type) aim

Attack	Anomalous causal subspaces
Neptune	1. $Num\_file\_creations \rightarrow Dst\_host\_srv\_serror\_rate$ 2. $Duration \rightarrow Dst\_host\_diff\_srv\_rate$
Back	1. $Flag \rightarrow Serror\_rate$ 2. $Protocol\_type, Dst\_host\_same\_srv\_rate \rightarrow Diff\_srv\_rate$
Satan	1. $Dst\_host\_srv\_serror\_rate, Srv\_diff\_host\_rate \rightarrow Dst\_host\_serror\_rate$ 2. $Protocol\_type, Dst\_host\_same\_srv\_rate \rightarrow Diff\_srv\_rate$
Smurf	1. $Diff\_srv\_rate, Dst\_host\_same\_srv\_rate \rightarrow Same\_srv\_rate$ 2. $Duration \rightarrow Dst\_host\_diff\_srv\_rate$
Ipsweep	1. $Duration \rightarrow Hot$ 2. $Loggedin, Src\_bytes \rightarrow Dst\_bytes$
Guesspwd	1. $Rerror\_rate, Dst\_host\_serror\_rate \rightarrow Dst\_host\_srv\_rerror\_rate$ 2. $Protocol\_type \rightarrow Srv\_count$
Rootkit	1. $Loggedin, Src\_bytes \rightarrow Dst\_bytes$ 2. $Flag \rightarrow Serror\_rate$
Land	1. $Service, Loggedin \rightarrow Srv\_diff\_host\_rate$ 2. $Num\_failed\_logins \rightarrow Num\_compromised$
Ftpwrite	1. $Srv\_count \rightarrow Dst\_host\_srv\_count$ 2. $Dst\_host\_count \rightarrow Duration$
Multihop	1. $Num\_file\_creations \rightarrow Num\_access\_files$ 2. $Protocol\_type, Dst\_host\_same\_srv\_rate \rightarrow Diff\_srv\_rate$
Phf	1. $Land, Num\_file\_creations \rightarrow Dst\_host\_srv\_diff\_host\_rate$ 2. $Flag \rightarrow Serror\_rate$
Spy	1. $Dst\_host\_count \rightarrow Dst\_host\_rerror\_rate$ 2. $Protocol\_type, Service, Flag \rightarrow Loggedin$
Perl	1. $Flag \rightarrow Serror\_rate$ 2. $Protocol\_type, Dst\_host\_same\_srv\_rate \rightarrow Diff\_srv\_rate$

Table 5.4: Domain specific anomalous causal subspaces discovered for various attacks in KDD Cup data set

for attributes, Numshell and Hot. Interestingly, few anomalous causal subspaces for attacks *Neptune*, *Smurf* and *Satan* were same as discovered by COM approach discussed in Chapter 4.

5.3.3 Discussion

Our experiments were designed to answer the following questions: (1) what are the key characteristics of an outlier which makes it really interesting? (2) is there any way which can ascertain quality of the reported outlier? and, (3) can we explain anomalies?.

The first question is addressed by the discussion we carried in the Introduction section and by the results given in Table 5.3. We discussed that outliers are not only "rare", "isolated" data points in the feature space, but are data points which violates the causal relationships. Distance (for example k^{th}-NN) and density (for example LOF) based approaches failed to give good accuracy in discovering anomalies. These approaches assume that outliers are isolated entities residing far away from their nearest neighbors or from a dense cluster. Such techniques consider each feature of the problem domain independent of each other. However, we claim that outliers are not just arbitrary "rare", "isolated" and low probable events, but are data points that suggest existence of unexpected causal structure which under domain knowledge is unlikely to appear.

In Figures 5.13 and 5.14 we present visualization of data points in training set, test and outlier set created for data sets Ecoli and Breast Cancer. We used PCA to reduce dimensionality of these data sets for the graphical representation. Figures strongly demonstrate the fact why k^{th}-NN and LOF techniques did not perform well on these data sets. Outliers for these data sets are present in the dense cluster of training set by which k^{th}-NN technique and LOF approaches failed to discover them. However, our algorithm worked well on these data sets since our approach works on the bases of relationship among attributes rather than treating them independent of each other. To prove this we show, example of outlier points in Ecoli data set which violated the causal semantic and was discovered successfully by our technique. In Figure 5.15, two dimensional visualization is shown for one causal subspace (Gvh $\rightarrow$ Mcg), refer Bayesian network on this data set in Figure 5.16. The figure explains the fact that outlier data points do not follow the trend of causal semantic between variables Gvh and Mcg and hence were detectable using our approach. Similar scenario is presented in Figure 5.16 where outlier data points in causal subspace (Gvh, Alm2 $\rightarrow$ Alm1) are shown. This implies for mining interesting anomalies it is important to consider the causal knowledge that exist among variables.

We addressed to the second question by proposing integration of domain knowledge using Bayesian networks in the discovering process. Domain knowledge helps quantify

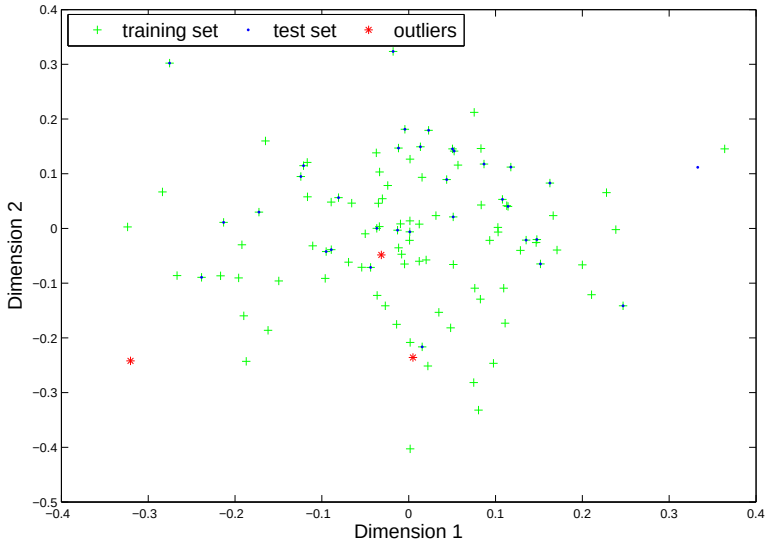


Figure 5.13: 2D visualization of Ecoli data set. Data points indicated by symbols +, . and * represents training data, test data and outliers respectively

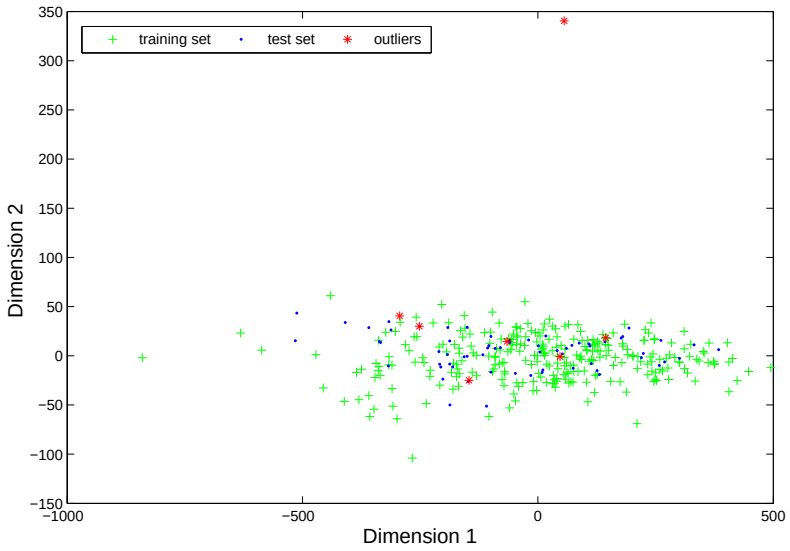


Figure 5.14: 2D visualization of Breast cancer data set. Data points indicated by symbols +, . and * represents training data, test data and outliers respectively

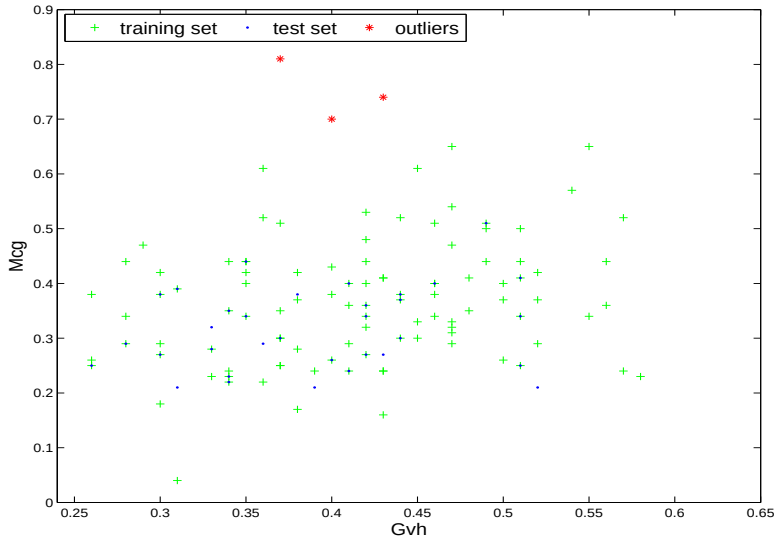


Figure 5.15: 2D visualization of Ecoli data set in causal subspace (Gvh $\rightarrow$ Mcg). Data points indicated by symbols +, . and * represents training data, test data and outliers respectively

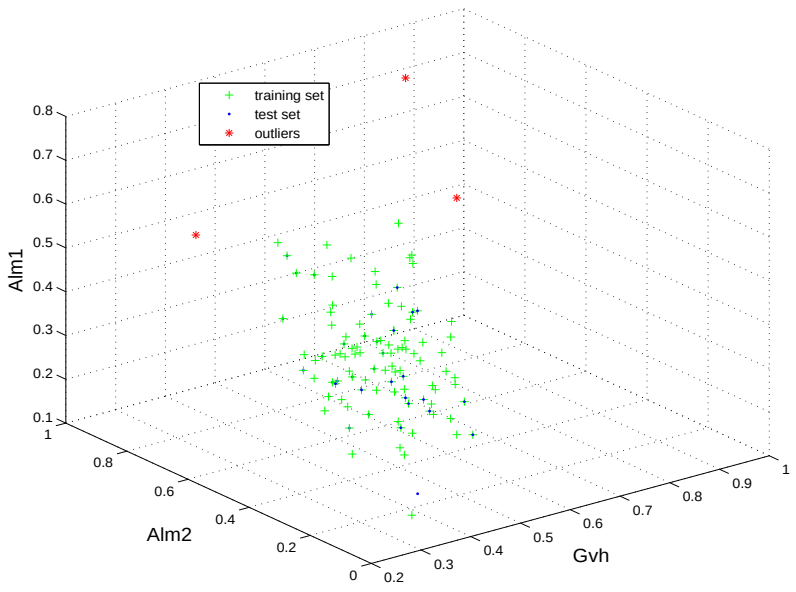


Figure 5.16: 3D visualization of Ecoli data set in causal subspace (Gvh, Alm2 $\rightarrow$ Alm1). Data points indicated by symbols +, . and * represents training data, test data and outliers respectively

the quality of the reported outlier. The results from Table 5.3 do not imply that our method can do best on an arbitrary data set. What the results do imply is that if the goal is to choose to the method which can claim mining of useful and interesting anomalies for the users then, most likely our approach is the best choice. Reason being that our approach mines anomalies based on the ground knowledge through which interestingness of the outlier can be reasoned.

Discovering anomalies from data is not itself a critical task. Rather more challenging is to provide insights on why discovered data point is an anomaly. An explanation capability is important to be integrated within discovery process for several reasons. *First*, explanations are needed to justify the recommendation. It increases the confidence and chances of acceptability of the technique in question. *Second*, through explanation facility, limitation of technique can be recognized. *Third*, an explanation provides insights of domain knowledge which can help develop and maintain a better environment. The explanation capability of our approach is addressed by the Table 5.4 where we show the important patterns of different attack types. Knowing these patterns, domain can help designing more robust environment against such malicious activities.

5.4 Summary and Conclusion

We proposed a measure to mine anomalies from data sets containing numerical or combination of discrete and numerical variables based on Gaussian and Hybrid Bayesian network respectively. Major objective of this work was to show that in order to mine interesting outliers, it is important to consider causality and correlation among attributes rather than treating them independently in the discovering process. Using causal reasoning, we exploited Bayesian structure and parametric information encoded in variables to uncover potentially strange and suspicions events. We have compared our approach with two classical outlier detection techniques, and have shown that our approach has both higher precision and recall. Another novel feature of our approach is that related to explanation of outliers. Through our approach, reported outlier can be explained on its anomalous nature which other outlier detection techniques may not.

Chapter 6

Conclusion and Future Work

6.1 Summary of The Research

Han and Kamber [36] defines outlier detection problem as follows: given a set of data points or objects, find a specific number of objects that are considerably dissimilar, exceptional and inconsistent with respect to the remaining data. Outliers are often interesting patterns that if found can raise alarms indicating that something unexpected is occurring in the process which has generated the data. As a consequence, outlier detection is one of the categories of knowledge discovery and an important research direction. In Chapter 2 we presented an overview of anomaly detection techniques which are related to or formed foundation of this thesis. We identified that existing studies on data mining has largely focused on the design of measures and algorithms to identify outliers in large and high dimensional categorical and numeric databases. However, not much stress has been given on the interestingness of the reported outlier.

We proposed, in order to discover "real" and "interesting" anomalies, integration of domain knowledge into discovery process is required. In this thesis, use of Bayesian networks is proposed to capture domain knowledge. Bayesian networks provided visualization of causal interactions among attributes that exist in the domain. And by exploiting these causal interactions, we were able to discover and explain anomalies. Under Bayesian setting, we defined interesting anomaly as those "data points which violate the the causal semantic captured via a Bayesian network".

We proposed solution techniques for anomaly detection in categorical, numerical or data sets containing mixture of categorical and numeric data values. These techniques

aimed for both *identification* and *explanatory* aspect of anomaly discovery.

In Chapter 4 we proposed a novel algorithm which combines the use of Bayesian network and probabilistic association rules to discover and explain anomalies in categorical data. The Bayesian network allowed us to organize information in order to capture both correlation and causality in the feature space, while the probabilistic association rules have a structure similar to association mining rules. In particular, we focused on two types of rules: (i) *low support & high confidence* and, (ii) *high support & low confidence*. New data points which satisfy either one of the two rules conditioned on the Bayesian network were the candidate anomalies. Extensive experiments performed on well-known benchmark data sets and demonstrated that our approach is able to identify anomalies in high precision and recall over existing traditional outlier detection techniques. Moreover, our approach can be used to discover contextual information from the mined anomalies, which other techniques often fail to do so.

In Chapter 5 we presented a measure to discover outliers in numerical data sets and data sets containing mixture of data types on the grounds of domain knowledge captured by a Gaussian Bayesian network and Hybrid Bayesian network respectively. We first built a Bayesian network depending upon type of data set given which encoded causal relationships between attributes and then identified those points as outliers which violate these causal relationships. Several experiments performed confirmed that outliers identified in this fashion are in some sense "genuine" as they reveal new information about the underlying data generating process.

In literature of outlier mining, outliers are often identified as data points which are "rare", "isolated", or "far away from their nearest neighbours". We demonstrated in Chapter 4 and 5 that meaningful outliers, i.e., outliers which perhaps encode important or new information are those which violate causal relationships. A critical analysis on distance based techniques was presented which highlights why distance based criteria may not be an accurate and effective technique to discover true outliers using real life examples and data sets. Also, we show why Bayesian approach could discover real outliers.

6.2 Future Work

This thesis has proposed a number of powerful techniques of mining interesting anomalies using domain knowledge captured by a Bayesian network. However, there are several promising research directions that can be extended from the work presented in this thesis.

- The suggested approaches for anomaly detection using Bayesian networks in this thesis are in particular designed for categorical, numerical and data sets containing mixture of data types. However, we can extend application of Bayesian networks for anomaly detection in temporal data. Murphy [55] defines a tool for modelling time-series data as Dynamic Bayesian Networks (DBNs). Using DBNs, inline with current research presented in this thesis, we see potential of exploring mining and explaining interesting anomalies in time series data.

- Techniques we proposed in this thesis were not applied on very high dimensional data sets. The maximum dimension we tried was 262. However, approaches we proposed in Chapter 4 and Chapter 5 can be extended for high dimensions. Recall from Chapter 4 and Chapter 5 that we considered each causal interaction independently in Bayesian networks which helped breaking the large, sparse network in small modules. Each module was then exploited using a simple form of causal inference, P(X | Pa(X)) where Pa(X) is set of parents of variable X which is always tractable. However, Bayesian structure and parameter learning for high dimensional data sets may be challenging. To overcome this problem, we suggest use of dimensionality reduction techniques such as Principal Component Analysis (PCA) [41] to retain meaningful features and then apply Bayesian structure learning algorithms [71; 70; 72; 77] to reveal Bayesian modelling.

Appendix A

Description of Bayesian networks

This appendix provides a comprehensive detail about the attributes used in Bayesian models shown in Chapter 4 and Chapter 5 of this thesis. The description of seven Bayesian networks namely: Zoo, Statlog, ChestClinic, Ecoli, Boston, NHL basket ball and KDD Cup intrusion detection are detailed in eight different sections below.

A.1 Description of Bayesian network built on Zoo Data set

Zoo data set from [4] contains 17 attributes describing different characteristics of animals such as number of fins and presence of hair on body. Bayesian structure learnt on this data set shown in Figure 4.7 of Chapter 4 revealed causal relationship among 11 attributes. Table A.1 describes these 11 feature names, meaning and their data type.

Feature	**Description**	**Data type**
Eggs	if animal lays egg	nominal
Fins	if animal has fins	nominal
Legs	number of legs	nominal
Hair	if animal has hair	nominal
Tail	if animal has tail	nominal
Aquatic	if its a aquatic animal	nominal
Domestic	if its a domestic animal	nominal
Predator	if animal naturally preys	nominal
Airborne	if animal can be transported	nominal
Toothed	if animal is flesh eater	nominal
Catsize	if animal belongs to a cat family	nominal

Table A.1: Summary of Zoo data set features

A.2 Description of Bayesian network built on Statlog Data set

It is a finical data set from [4] which describes important attributes which are assessed before granting credit to a person. It contains 21 attributes in total. Bayesian structure learnt on this data set shown in Figure 4.9 of Chapter 4 revealed causal relationship among 17 attributes. Table A.2 describes these 4 feature names, meaning and their data types.

Feature	**Description**	**Data type**
Employment	if person is employed	nominal
Job	Job type of the person	nominal
Credit history	credit history of the person	nominal
Credit amount	amount of credit asked	nominal
Own telephone	if person owns telephone	nominal
Property magnitude	property owned by the person	nominal
Residence since	present property since	nominal
Other parties	other debtors/ guarantors	nominal
Housing	if person has own or rented house	nominal
Foreign worker	if person works in foreign	nominal
Duration	duration in month	nominal
Checking status	status of existing checking account	nominal
Num dependents	number of dependents	nominal
Age	age in years	nominal
Existing credits	number of existing credits in bank	nominal
Personal status	if person is married or single	nominal
Personal status	if person is married or single	nominal

Table A.2: Summary of Statlog data set features

A.3 Description of Nodes in ChestClinic Bayesian network

A simple Bayesian network proposed by [50] is available on Netica Bayesian repository [23]. It is useful in diagnosing patients arriving at a clinic. The Bayesian network is shown in Figure 4.11 of Chapter 4 of this thesis. Table A.3 describes 8 feature names, meaning and their data types.

Feature	**Description**	**Data type**
Visit to Asia	if patient has recently visited Asia	nominal
Smoking	if person smokes	nominal
Tuberculosis	if person suffers from tuberculosis	nominal
Lung cancer	if person suffers from lung cancer	nominal
Cancer	if person has cancer	nominal
Bronchitis	if person suffers from Bronchitis	nominal
Dyspnea	if person has Dyspnea	nominal
X-ray	X ray report of the person	nominal

Table A.3: Summary of nodes in Chestclinic Bayesian network

A.4 Description of Bayesian network built on Ecoli Data set

Ecoli data set from [4] contains 8 attributes describing protein localization sites. Bayesian structure learnt on this data set shown in Figure 5.7 of Chapter 5 revealed causal relationship among 4 attributes. Table A.4 describes these 4 feature names, meaning and their data types.

Feature	Description	Data type
Gvh	Von Heijne's method for signal sequence recognition	real
Mcg	McGeoch's method for signal sequence recognition	real
Alm1	score of ALOM membrane spanning region prediction recognition	real
Alm2	score of ALOM program after excluding putative cleavable signal regions from the sequence	real

Table A.4: Summary of Ecoli data set features

A.5 Description of Bayesian network built on Boston Data set

This data set concerns housing values in suburbs of Boston taken from [4]. It contains 14 features. The Bayesian network shown in Figure 5.8 of Chapter 5 on this data set resulted in a model containing 12 attributes. Table A.5 describes these 12 feature names, meaning and their data types.

Feature	**Description**	**Data type**
CRIM	per captia crime rate by town	real
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.	real
INDUS	proportion of non-retail business acres per town	real
NOX	nitric oxides concentration (parts per 10 million)	real
RM	average number of rooms per dwelling	real
AGE	proportion of owner-occupied units built prior 1940	real
DIS	weighted distances to five Boston employment centres	real
RAD	index of accessibility to radial highways	real
TAX	full-value property-tax rate per $10,000	real
PTRATIO	pupil-teacher ratio by town	real
B	$1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town	real
LSTAT	percentage of lower status population	real

Table A.5: Summary of Boston data set features

A.6 Description of Bayesian network built on NHL basket ball Data set

This data set contains information of all the players in the famous basketball league in the US in year 1997-1998. The Bayesian model on this data set is show in Figure 5.9 of Chapter 5 of this thesis. Table A.6 describes 6 feature names, meaning and their data types.

Feature	**Description**	**Data type**
Position	position of the player	nominal
Reb-p-g	rebounds per game (usually an indicator of defensive ability)	real
Asts-p-g	assists per game	real
Pts-p-g	points per game	real
Blk-p-g	blocks per game	real
Stl-p-g	steals per game	real

Table A.6: Summary of NHL basket ball data set features

A.7 Description of Bayesian network built on KDD Cup Intrusion Detection Data set

KDD data set from [4] is one of the few publicly available data sets for network-based anomaly detection systems. It contains 42 (including class label) attributes. The Bayesian network is shown in Figure 4.10 and Figure 5.10 of Chapter 4 and Chapter 5 respectively of this thesis. Table A.7 describes these features, meaning and their data types.

Feature	**Description**	**Data type**
Duration	duration of the connection	continuous
Protocol type	type of protocol, e.g., tcp, udp etc.	nominal
Service	network service on destination	nominal
Src_bytes	number of data bytes from sent	continuous
Dst_bytes	number of data bytes received	continuous
Flag	normal or error status of the connection	nominal
Land	1 if connection is from/to the same host/port;0 otherwise	nominal

Wrong_fragment	number of wrong fragments	continuous
Urgent	number of urgent packets	continuous
Count	number of connections to the same host as the current connections in the past two seconds	continuous
Serror_rate	% of same host connections that have "SYN" errors	continuous
Rerror_rate	% of same host connections that have "REJ" errors	continuous
Same_srv_rate	% of same host connections to the same services	continuous
Diff_srv_rate	% of same host connections to different services	continuous
Srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
Srv_serror_rate	% of same service connections that have "SYN" errors	continuous

Srv_rerror_rate	% of same service connections that have "REJ" errors	continuous
Srv_diff_host_rate	% of same service connections to different hosts	continuous
Hot	hot indicators	continuous
Num_failed_logins	number of failed login attempts	continuous
Loggedin	1 if successful logged in;0 otherwise	nominal
Num_compromised	number of the compromised states on the destination host	continuous
Root_shell	1 if root shell is obtained;0 otherwise	nominal
Su_attempted	1 if "su root" command attempted; 0 otherwise	nominal
Num_root	number of "root" accesses	continuous
Num_file_creations	number of file creations	continuous

Num_access_files	number of operations on access control files	continuous
Num_outbound_cmds	number of outbound commands in an ftp session	continuous
Is_guest_login	1 if the login belongs to the "guest" list; 0 otherwise an ftp session	nominal
Is_host_login	1 if the login belongs to the "host" list; 0 otherwise	nominal
Same_srv_rate	% of same host connections to the same services	continuous
Diff_srv_rate	% of same host connections to different services	continuous
Srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
Srv_serror_rate	% of same service connections that have "SYN" errors	continuous
Srv_rerror_rate	% of same service connections that have "REJ" errors	continuous
Srv_diff_host_rate	% of same service connections to different hosts	continuous

Dst_host_count	number of connections to the same host in the past 100 connections	continuous
Dst_host_serror_rate	% of connections that have "SYN" errors	continuous
Dst_host_rerror_rate	% of connections that have "REJ" errors	continuous
Dst_host_same_srv_rate	% of connections to same service	continuous
Dst_host_diff_srv_rate	% of same host connections to different services	continuous
Dst_host_srv_count	number of connections to the same service in the past 100 connections	continuous
Dst_host_srv_serror_rate	% of same service connections that have "SYN" errors	continuous
Dst_host_srv_rerror_rate	% of same service connections that have "REJ" errors	continuous
Dst_host_srv_diff_host_rate	% of same service connections to different hosts	continuous
Dst_host_same_src_port_rate	% of connections from the same source port	continuous

Table A.7: Summary of KDD Cup intrusion detection data set features

A.8 Summary of Attacks in KDD Cup Intrusion Detection Data set

Each record in the data set is labeled either as normal or as an attack. There are 22 different attacks present in this data set. All these types are classified into four major categories described below.

Probe: The probe attacks are carried out usually for reconnaissance purposes. For instance, a network can be probed to gather information about the types and number of computers connected to a network, a host can be probed to find out the types of installed services or the types of user accounts configured on it.

User to Root (U2R): In these types of attacks the aim of the attacker is to gain illegal access to the super-user or administrative account privileges to abuse resources or to get access to classified documents.

Denial Of Service (DOS): DOS attacks are targeted at disrupting a normal service or completely making it unavailable for normal usage.

Remote to Local (R2L): These attacks provide illegal access to an attacker, who has access to send packets to a remote network, to the local users accounts.

Table A.8 lists attack types in KDD Cup intrusion detection data set, their categorisation and number of samples present in data set.

Attack type	**Category**	**Samples**
Satan	Probe	906
Ipsweep	Probe	651
Portsweep	Probe	416
Nmap	Probe	158
Buffer_overflow	U2R	30
Rootkit	U2R	10
Loadmodule	U2R	9
Perl	U2R	3
Neptune	DOS	51820
Back	DOS	968
Teardrop	DOS	918
Smurf	DOS	641
Pod	DOS	206
Land	DOS	19
Warezclient	R2L	893
Guess_pwd	R2L	52
Ftp_write	R2L	8
Multihop	R2L	7
Phy	R2L	4
Spy	R2L	2
Imap	R2L	0
Warezmaster	R2L	0

Table A.8: Categories of attacks and their samples present in KDD Cup intrusion detection data set

Bibliography

[1] Database basketball. http://www.databasebasketball.com.

[2] Statlib cmu. http://lib.stat.cmu.edu/.

[3] Samiam: Bayesian sensitivity, modeling, inference and analysis software. URL http://reasoning.cs.ucla.edu/samiam/.

[4] Uci machine learning repository. URL http://archive.ics.uci.edu/ml.

[5] S. Andreassen, Marianne W., Bjorn F., and Stig K. A. Munin - A causal probabilistic network for interpretation of electromyographic findings. In *Proc. of the 10th International Joint Conf. on Artificial Intelligence*, pages 366–372, 1987.

[6] B-Course. A web-based data analysis tool for bayesian modeling. URL http://b-course.cs.helsinki.fi/obc/.

[7] Sakshi Babbar and Sanjay Chawla. On Bayesian Networks and Outlier Detection. In *Proceedings of the 16th International Conference on Management of Data*, pages 125–137, 2010.

[8] Sakshi Babbar and Sanjay Chawla. Mining Causal Outliers Using Gaussian Bayesian Networks. In *Proceedings of the IEEE 24th International Conference on Tools with Artificial Intelligence*, pages 97–104, 2012.

[9] Sakshi Babbar, Didi Surian, and Sanjay Chawla. A Causal Approach for Mining Interesting Anomalies. In *Proceedings of the 26th Canadian Conference on Artificial Intelligence*, pages 226–232, 2013.

[10] D. Barbara, N. Wu, and S. Jajodia. Detecting Novel Network Intrusions using Bayes Estimators. In *Proc. SIAM Intl. Conf. Data Mining*, 2001.

[11] Vic Barnett and Toby Lewis. *Outliers in Statistical Data*. Wiley Series in Probability & Statistics. Wiley, 1994.

[12] Stephen D. Bay and Mark Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proc. of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 29–38, 2003.

[13] R. E Bellman. *Adaptive control processes: A Guided Tour*. Princeton University Press, 1961.

[14] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[15] A. Bovas and C. Alice. Outlier detection and time series modeling. *Technometrics*, 31:241–248, 1989.

[16] A. Bovas and E. P. B. George. Bayesian analysis of some outlier problems in time series. *Biometrika*, 66:229–236, 1979.

[17] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying Density-Based Local Outliers. In *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, pages 93–104. ACM, 2000.

[18] Re Bronstein, Joydip Das, Marsha Duro, Rich Friedrich, Gary Kleyner, Martin Mueller, Sharad Singhal, Ira Cohen, G. Kleyner, M. Mueller, S. Singhal, and I. Cohen. Self-aware services: Using bayesian networks for detecting anomalies in internet-based services. In *In International Symposium on Integrated Network Management*, pages 623–638, 2001.

[19] Antonio Cansado and Alvaro Soto. UNSUPERVISED ANOMALY DETECTION IN LARGE DATABASES USING BAYESIAN NETWORKS. *Appl. Artif. Intell.*, 22:309–330, 2008.

[20] Sanjay Chawla and Pei Sun. SLOM: a new measure for local spatial outliers. *Knowledge Information System*, 9:412–429, 2006.

[21] Seong-Pyo Cheon, Sungshin Kim, So Young Lee, and Chong-Bum Lee. Bayesian networks based rare event prediction with sensor data. *Knowl.-Based Syst.*, 22: 336–343, 2009.

[22] Gregory F. Cooper and Tom Dietterich. A Bayesian method for the induction of probabilistic networks from data. In *Machine Learning*, pages 309–347, 1992.

[23] Norsys Software Corp. Norsys software corporation: Bayes net library. URL https://www.norsys.com/netlibrary/index.htm.

[24] Norsys Siftware Corporation. Bayesian network development software. URL https://www.norsys.com/.

[25] V. Reddy D. Janakiram and A. Kumar. Outlier Detection in Wireless Sensor Networks using Bayesian Belief Networks. In *Proceedings of the 1st International Conference on Communication System Software and Middleware*, pages 1–6, 2006.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39:1–38, 1977.

[27] Christopher P. Diehl and John B. Hampshire Ii. Real-time Object Classification and Novelty Detection for Collaborative Video Surveillance. In *Proceedings of the International Joint Conference on Neural Networks*, pages 2620–2625, 2002.

[28] J R Dorronsoro, F Ginel, C Sgnchez, and C S Cruz. Neural fraud detection in credit card operations. *IEEE Transactions on Neural Networks*, 8:827–834.

[29] Hugin Expert. Bayesian network and influence diagrams development software. URL http://www.hugin.com/.

[30] E.M. Ferragut, D.M. Darmon, C.A. Shue, and S. Kelley. Automatic construction of anomaly detectors from graphical models. In *Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on*, pages 9–16, 2011.

[31] Nir Friedman, Iftach Nachman, and Dana Peér. Learning bayesian network structure from massive datasets: the "sparse candidate" algorithm. In *Proceedings of*

the Fifteenth conference on Uncertainty in artificial intelligence, pages 206–215, 1999.

[32] Bottcher S. G. and Dethlefsen C. Deal: A package for learning bayesian networks. URL http://cran.r-project.org/.

[33] GeNIE and SMILE. Bayesian network and influence diagrams development software, . URL http://genie.sis.pitt.edu/.

[34] GeNIE and SMILE. Bayesian repository, . URL http://genie.sis.pitt.edu/index.php/network-repository.

[35] T. L. Griffiths and J. B. Tenenbaum. From mere coincidences to meaningful discoveries. *Cognition*, 103:180–226, 2007.

[36] Jiawei Han and Micheline Kamber. *Data mining : concepts and techniques*. Morgan Kaufmann, 2005.

[37] D. Hawkins. *Identification of Outliers*. Chapman and Hall, London, 1980.

[38] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3): 197–243, September 1995. Available as Technical Report MSR-TR-94-09.

[39] David E. Heckerman, E. J. Horvitz, and B. N. Nathwani. Toward Normative Expert Systems: Part I The Pathfinder Project. pages 204–28, 1993.

[40] FinnV. Jensen, SørenH. Aldenryd, and KlausB. Jensen. Sensitivity analysis in bayesian networks. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 946, pages 243–250. 1995.

[41] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, NY, second edition, 2002.

[42] Murphy K. Bayes net toolbox for matlab. URL http://bnt.googlecode.com/svn/trunk/docs/usage.html.

[43] J. H. Kim and J. Pearl. CONVINCE: a conversational inference consolidation engine. *IEEE Trans. Syst. Man Cybern.*, 17:120–132, 1987.

[44] U. B. Kjaerulff and A. L. Madsen. *Probabilistic Networks- An Introduction to Bayesian Networks and Influence Diagrams*. 2005. URL http://people.cs.aau.dk/~uk/papers/pgm-book-I-05.pdf.

[45] Edwin M. Knorr and Raymond T. Ng. Algorithms for Mining Distance-Based Outliers in Large Datasets. In *The 24rd International Conference on Very Large Data Bases*, pages 392–403, 1998.

[46] Edwin M. Knorr and Raymond T. Ng. Finding Intensional Knowledge of Distance-Based Outliers. In *VLDB*, pages 211–222, 1999.

[47] Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8:237–253, 2000.

[48] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT press, 2009.

[49] Wai Lam and Fahiem Bacchus. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, 10:269–293, 1994.

[50] S. L. Lauritzen and D. J. Spiegelhalter. Readings in uncertain reasoning. chapter Local computations with probabilities on graphical structures and their application to expert systems, pages 415–448. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-125-2.

[51] Bayes Server Limited. Bayesian sensitivity, modeling, inference and analysis software. URL http://www.bayesserver.com/.

[52] C. Mahalanobis, P. On the generalised distance in statistics. In *Proceedings National Institute of Science*, pages 49–55, 1936.

[53] Dimitris Margaritis. *Learning Bayesian Network Model Structure from Data*. PhD thesis, Carnegie Mellon University, Pittsburgh, 2003.

[54] Markos Markou and Sameer Singh. Novelty Detection: A Review - Part 1: Statistical Approaches. *Signal Processing*, 83:2003, 2003.

[55] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning* . PhD thesis, University of California, Berkeley, 2002.

[56] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

[57] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[58] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

[59] S. Pei. *Outlier Detection In High Dimensional, Spatial And Sequential Data Sets*. PhD thesis, The University of Sydney, 2006.

[60] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 427–438, New York, NY, USA, 2000. ACM. doi: http://doi.acm.org/10.1145/342009.335437.

[61] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6:461–464, 1978.

[62] Marco Scutari. bnlearn: Bayesian network learning and inference, . URL http://www.bnlearn.com/.

[63] Marco Scutari. Bayesian network repository, . URL http://www.bnlearn.com/.

[64] Abdallah Abbey Sebyala, Temitope Olukemi, Lionel Sacks, and Lionel Sacks. Active Platform Security through Intrusion Detection Using Naive Bayesian Network For Anomaly Detection. In *Proceedings of London communications symposium*, 2002.

[65] W. A. Shewhart. *Economic Control of Quality of Manufactured Product*. D. Van Nostrand Company, New York, 1931.

[66] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering*, 19 (5):631–645, 2007.

[67] Peter Spirtes, Clark Glymour, and Richard Scheines. Causation, prediction, and search. 1993.

[68] Pei Sun and Sanjay Chawla. On Local Spatial Outliers. In *Proc. of the Fourth IEEE International Conference on Data Mining*, pages 209–216. IEEE Computer Society, 2004.

[69] Pei Sun, Sanjay Chawla, and Bavani Arunasalam. Mining for Outliers in Sequential Databases. In *SDM*, 2006.

[70] I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Algorithms for Large Scale Markov Blanket Discovery. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, pages 376–381. AAAI Press, 2003.

[71] I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 673–678. ACM, 2003.

[72] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Machine Learning*, 65(1):31–78, 2006.

[73] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1978.

[74] C. Varun, B. Arindam, and K. Vipin. Anomaly detection: A survey. *ACM Computing Survey*, 41:15:1–15:58, July 2009. ISSN 0360-0300.

[75] W. Wong, A. Moore, G. Cooper, and M. Wagner. Bayesian Network Anomaly Pattern Detection for Disease Outbreaks. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 808–815, Menlo Park, California, August 2003. AAAI Press.

[76] Liang Xiong, Barnabás Póczos, Jeff G. Schneider, Andrew Connolly, and Jake VanderPlas. Hierarchical probabilistic models for group anomaly detection. *Journal of Machine Learning Research - Proceedings Track*, 15:789–797, 2011.

[77] S. Yaramakala and D. Margaritis. Speculative Markov Blanket Discovery for Optimal Feature Selection. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 809–812. IEEE Computer Society, 2005.

[78] Nong Ye and Mingming Xu. Probabilistic Networks with Undirected Links for Anomaly Detection. In *Proceedings of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, pages 175–179, 2000.