August 2012

An Efficient Methodology for Learning Bayesian Networks

Emmanuel Owusu Asante-Asamani
University of Wisconsin-Milwaukee

AN EFFICIENT METHODOLOGY FOR LEARNING BAYESIAN NETWORKS

by

Emmanuel Asante-Asamani

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Mathematics

at

University of Wisconsin Milwaukee

August 2012

ABSTRACT

AN EFFICIENT METHODOLOGY FOR LEARNING BAYESIAN NETWORKS

by

Emmanuel Asante-Asamani

The University of Wisconsin Milwaukee, 2012
Under the Supervision of Professor Istvan Lauko

Statistics from the National Cancer Institute indicate that 1 in 8 women will develop Breast cancer in their lifetime. Researchers have developed numerous statistical models to predict breast cancer risk however physicians are hesitant to use these models because of disparities in the predictions they produce. In an effort to reduce these disparities, we use Bayesian networks to capture the joint distribution of risk factors, and simulate artificial patient populations *(clinical avatars)* for interrogating the existing risk prediction models. The challenge in this effort has been to produce a Bayesian network whose dependencies agree with literature and are good estimates of the joint distribution of risk factors. In this work, we propose a methodology for learning Bayesian networks that uses prior knowledge to guide a collection of search algorithms in identifying an optimum structure. Using data from the breast cancer surveillance consortium we have shown that our methodology produces a Bayesian network with consistent dependencies and a better estimate of the distribution of risk factors compared with existing methods.

TABLE OF CONTENTS

LIST OF FIGURES vii

LIST OF TABLES ix

ACKNOWLEDGEMENTS x

Chapter 1: Introduction 1

1.1 Background 1

1.2 Risk prediction Models 3

1.3 Reducing Health disparities by simulated populations 4

Chapter 2: Theoretical Background to Bayesian Networks 7

2.1 Estimating Joint Probability Distribution 7

2.1.1 DAG's and Probability Distribution 8

2.1.2 The Markov Condition 9

2.1.3 Faithfulness and Minimality Condition 11

2.1.4 D Separation 13

2.2 Learning Bayesian Networks 18

2.2.1 Bayesian Learning 20

2.2.2 Constrained Based Learning 25

2.3 Model selection 30

2.3.1 Bayesian Scoring Metric (BDe) 30

2.3.2 Kullback-Leibler Distance (KLD) 34

2.3.3 Mutual information 35
2.3.4 Bayesian Information Criterion 36
2.4 Parameter Estimation 37
2.4.1 Maximum Likelihood (ML) Estimation 37
2.4.2 Maximum a posterior Estimation 38
Chapter 3: Iterative Knowledge Guided Search 41
3.1 Preprocessing of Data 42
3.2 Learning 43
3.2.1 Training 43
3.2.2 Performance Evaluation 46
3.3 Validation 46
Chapter 4: Application to Breast Cancer Risk Prediction 47
4.1 Data Description and Preprocessing 47
4.2 Learning 52
4.2.1 Training 52
4.2.2 Performance Evaluation 56
4.3 Validation 57
Chapter 5: Conclusion and Recommendation 61
5.1 Conclusions 61
5.2 Recommendations 62

Bibliography 69

LIST OF FIGURES

Figure 1-1: Normal breast with non–invasive ductal carcinoma in situ (DCIS) in an enlarged cross–section of the duct 2

Figure 2-1: Direct Dependence 8

Figure 2-2: Directed Acyclic Graph illustrating Markov Condition 10

Figure 2-3: Illustrating Minimality condition 12

Figure 2-4: Illustrating D-Separation 14

Figure 2-5: Blocked and Active paths illustrating d-separation 16

Figure 2-6: Illustrating d--separation by more than one node 17

Figure 2-7: A Bayesian network constructed from causal knowledge 19

Figure 2-8: Search States of a Bayesian Learning Algorithm 22

Figure 2-9: Gold Standard Bayesian Network 26

Figure 2-10: Complete Undirected Graph 26

Figure 2-11: Skeleton of Gold Standard Network 27

Figure 2-12: Meeks Orientation Rules 29

Figure 2-13: Orienting DAG's using Meeks rules, (a) Orienting colliders; 29

Figure 3-1: Workflow for Iterative Knowledge Guided Search 42

Figure 4-1: Multiple Imputations with Chained Equations 49

Figure 4-2: Comparison of sampled data with original data 49

Figure 4-3: Comparison of sampled data with imputed data 51

Figure 4-4: Entering Prior Knowledge into TETRAD 53

Figure 4-5: Knowledge updates per Iteration 54

Figure 4-6: Best Performing DAG's 56

Figure 4-7: Mined Model (a) and IKGS model (b) 58

Figure 4-8: Reduced mined model 59

LIST OF TABLES

Table 3-1 : Summary of Search Algorithms 45

Table4-1: Description of Variable 50

Table 4-2 : Results of metrics for each iteration 55

Table 4-3: Comparing metrics of Mined model and IKGS model 60

ACKNOWLEDGEMENTS

I would like to thank my committee members: Bruce Wade, Peter Tonellato, Jugal Ghorai and Istvan Lauko for their support and guidance. Matthew Crawford, Michiyo Yamada, Omid Ghiasvand for contributing valuable ideas towards this modeling effort. Finally, my family and friends for their encouragement.

Chapter 1: Introduction

1.1 Background to Breast Cancer

Breast Cancer is a cancer that is initiated from the tissues of the breast. There are two main types: Ductal carcinoma, which starts in the milk ducts and Lobular carcinoma which starts in the lobules. The most common form of breast cancer is ductal carcinoma. The disease may be invasive, which typically describes the stage where the cancer has spread to nearby tissues, or non invasive *(in situ)* which is when the disease is contained in a particular breast tissue. Breast cancer may be classified as being in stage I, II, III or IV. Usually stage I-III can be treated through procedures such as lumpectomy, mastectomy, hormone therapy, or chemotherapy to remove the cancerous cells. Stage IV cancer's are generally incurable and can only be managed to prolong life.

Statistics from the national cancer center indicates that 1 in every 8 women born in the US will develop breast cancer in their lifetime (Institute 2010). This makes it imperative for every woman to regularly examine herself for any symptoms of the disease and have it treated early before it becomes malignant. Common symptoms of breast cancer include: breast lumps, change in size, shape or feel of the breasts, unusual fluid coming from the nipple.

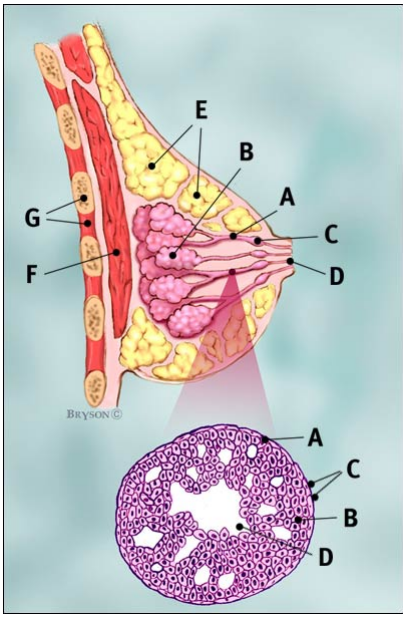


Figure 1-1: Normal breast with non–invasive ductal carcinoma in situ (DCIS) in an enlarged cross–section of the duct

An even more important preventive measure is for every woman to know her risk of getting breast cancer so physicians can perform regular examinations to detect any onset of the disease. There are a number of factors that tend to increase a woman's risk for breast cancer namely age, family history of breast cancer, genes, menstrual cycle, alcohol use, childbirth, hormone replacement therapy (HRT) and radiation. Typically the disease is more prevalent in women over the age of 50 years and those who have close relatives with breast cancer (reference). Women with defective BRCA1 and BRCA2 genes are also at risk of getting breast cancer. These genes usually produce proteins that prevent

cancer. Any mutations can produce a counter effect. It has been reported that women who got their periods early (<12 years) or experienced a late menopause (>50 years) have an increased risk for breast cancer. Research also shows that having more than 1-2 glasses of alcohol a day may increase the incidence of breast cancer. Women who have received some form of hormone replacement therapy with estrogen also have an increased risk for breast cancer. Exposure to radiation around the chest area may also lead to higher risk for breast cancer.

1.2 Risk prediction Models

A number of statistical models have been developed to predict a woman's risk for breast cancer. Gail in 1989 produced a model that gives a five year risk for breast cancer based on age at menarche, age at first live birth, number of previous biopsies, and number of first-degree relatives with breast cancer (Gail 1989). In 1999 he formulated an improved model by including history of atypical hyperplasia and in 2007 extended his model to an African American population. Other models have resulted from some modification of the Gail model either by including more risk factors or extending to a different population. For example the Tice model (Jeffrey A. Tice 2008) developed in 2008 included breast density and race into the Gail 1999 model and extended to a US mixed population. Chlebowski (Richard J Santeen 2007) also added alcohol, bmi, hrt, breast feeding, physical activity, parity and smoker to the Gail 1999 model and also extended to a US

mixed population. Similar models have been developed for Japanese, Korean, Italian and European mixed populations.

1.3 Reducing Health disparities by simulated populations

There are several risk prediction models out there, each developed with a different study population and data set. For the physician at the point of care, it is important to decide which model is suitable for a patient's unique characteristics. Unfortunately, the lack of a comprehensive assessment of these predictive models makes that task difficult, occasionally resulting in inaccurate risk predictions. The center for Biomedical Informatics (HMS-at Harvard Medical School) and The Laboratory for Public Health Informatics and Genomics (LPHIG – at UWM) have begun efforts to reduce this disparity by interrogating currently existing risk prediction models to identify and document their strengths and weaknesses. The project begun with an extensive review of all currently existing risk prediction algorithms and the construction of a pedigree to illustrate the relationships between them. The project is currently in its second phase where a Bayesian network model describing the dependencies between the risk factors is required to simulate artificial patient populations (*clinical avatars*) for the interrogation of the risk prediction models.

Bayesian networks have become the tool of choice by most researchers for knowledge discovery because of their facility in approximating complex multivariable distributions

and incorporating prior domain knowledge. Knowledge obtained from Bayesian networks have been used in a wide variety of applications. For instance, high level biological knowledge obtained from gene ontologies have been incorporated into Bayesian networks trained on protein interaction data for diagnostic reasoning and prediction of protein function. (Jung Hun Oh 2011) also used Bayesian networks to predict local failure in lung cancer and recorded significant improvement in their predictions compared with standard dose-volume models. Nurse researchers are now able to incorporate both clinical and theoretical knowledge in mining very large hospital information data bases using Bayesian networks (Sun-Mi Lee 2003). Knowledge from Bayesian network have also been used in facilitating secondary use of EMR data for predicting study outcomes, conducting retrospective studies and simulating clinical trials.

The literature is filled with a plethora of algorithms for training Bayesian networks (David Heckerman 1995; Peter Spirtes 2000; Chickering 2002), but as pointed out by (Guoliang LI 2007) most of the learned networks produce edges which may be inconsistent with domain knowledge. The performance of Bayesian networks seems to rely heavily on characteristics of the problem domain making it difficult to rank one algorithm as preferable to others(Mozaherul Hoque Abul Hasanat 2010).

In this work we propose a methodology for training Bayesian networks that harnesses the strengths of already existing algorithms to produce Bayesian networks which offer

improved estimation of the distribution of random variables. We show that our methodology when applied to modeling breast cancer risk produces edges consistent with literature, making it ideal for simulating clinical avatars.

Chapter 2: Theoretical Background to Bayesian Networks

2.1 Estimating Joint Probability Distribution

Consider the random variables X, Y, Z, W, T for which we would like to obtain their joint probability distribution, $P(X, Y, Z, W, T)$. By the chain rule of probability we can express the joint distribution in the form,

$$P(X, Y, Z, W, T) = P(X)P(Y \mid X)P(Z \mid X, Y)P(W \mid X, Y, Z)P(T \mid X, Y, Z, W, T) \quad (2.1.1)$$

What remains is be to estimate the conditional distribution of each of the terms on the RHS of (2.1.1). Suppose for simplicity and convenience of illustration that X, Y, Z, W, T are discrete binary random variables, then the conditional distributions would be relative frequencies of the different values of each variable. A total of 31 free parameters would need to be estimated to fully specify the joint distribution. The breakdown is as follows, $[P(X) - 1;\ P(Y|Z) - 2;\ P(Z|X, Y) - 4; P(W|X, Y, Z) - 8; P(T|X, Y, X, W, T) - 16]$. Now suppose we knew the following conditional independence facts about the random variables: $T \perp X, Y|Z, W;\ Z \perp W|X, Y;\ W \perp Z, X|Y$ and finally $X \perp Y$, then the joint distribution (2.1.1) could be simplified to the form

$$P(X, Y, Z, W, T) = P(X)P(Y)P(Z \mid X, Y)P(W \mid Y)P(T \mid Z, W) \quad (2.1.2)$$

which reduces the number of free parameters to be estimated to 10. Thus by applying knowledge of conditional independence facts about the random variables we are able to

reduce the number of parameters by 21. This might not appear to present much savings in computation of the joint distribution however if instead of five binary variables we now have twenty multinomial variables, it becomes clear the significant amount of computational time saved, making the estimation process more tractable.

2.1.1 DAG's and Probability Distribution

A Bayesian network is a probability graphical model which encodes knowledge of the conditional independence facts among a set of random variables. It consists of nodes which represent random variables and edges which indicate the independence relations between them. The presence of an edge between two nodes is an indication that the two random variables are directly dependent. The absence of an edge on the other hand is an indication of conditional independence. All edges in a Bayesian network are directed (i.e. they have an arrow head at one end which indicates the direction of the dependency). If two variables X and Y have a directed edge between them as illustrated in Figure 2-1 then the probability distribution over Y is dependent on X.The variable X is termed a *parent* of Y and Y a *child* of X.

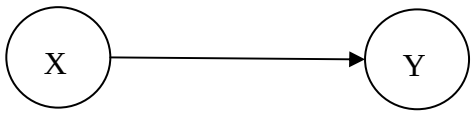


Figure 2-1: Direct Dependence

in a particular direction then the path is said to be a *directed path* otherwise it is

undirected. For a sequence of nodes $XYZWT$ on a directed path such that the path is out of X and into T, the nodes XYZ are referred to as *ancestors* of T and T a *descendant* of X. If there is no directed path that starts from one node and ends in the same node, the graph is said to be *acyclic*. Bayesian networks are typically directed acyclic graphs.

Given a Bayesian network whose structure correctly represents the conditional independence relationships among random variables, we are able to simplify the factorization of the joint distribution and conveniently estimate its parameters.

2.1.2 The Markov Condition

Suppose we have a Bayesian network whose structure is an accurate representation of the conditional independence relationships among a set of random variables, we are able, by mean of the Markov Condition, to extract all the conditional independence facts necessary to simplify the representation of the joint distribution. The Condition states:

> *Definition: (Markov Condition)*
>
> A directed acyclic graph (DAG) and a probability distribution satisfy the Markov condition if every node in the graph is conditionally independent of all its non-descendants given its parents (Pa_j) i.e
>
> $$P(X_j \mid X_1 \mathrm{L}\ X_n, Pa_j) = P(X_j \mid Pa_j)$$
>
> *Where $X_1 \cdots X_n$ are non-descendants of X_j*

Therefore if a Bayesian network and probability distribution satisfy the Markov Condition, then by identifying the parents of each node in the graph the conditional

independence relationships necessary to reduce the factorization of the joint distribution can be extracted.

Consider the structure in Figure 2-2, by the chain rule of probability the joint distribution can be factorized as follows:

$$P(X,Y,Z,W,T) = P(X)P(Y \mid X)P(Z \mid Y,X)P(W \mid X,Y,Z)P(T \mid X,Y,Z,W) \qquad (2.1.3)$$

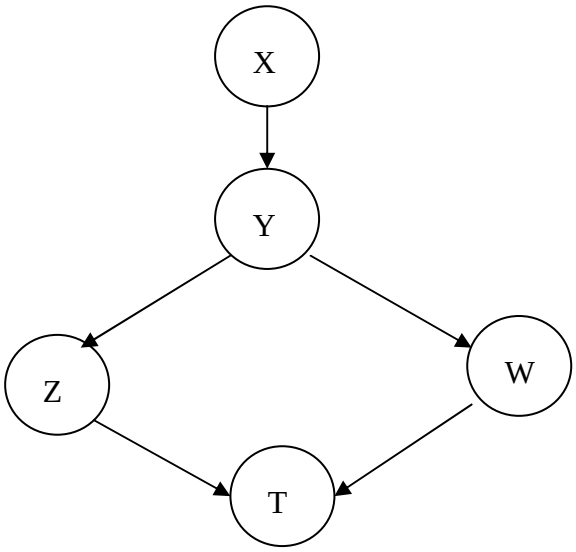


Figure 2-2: Directed Acyclic Graph illustrating Markov Condition

Now by applying the Markov condition the joint distribution (2.1.3) simplifies to,

$$P(X,Y,Z,W,T) = P(X)P(Y \mid X)P(Z \mid Y)P(W \mid Y)P(T \mid Z). \qquad (2.1.4)$$

From which we can estimate the parameters of the distribution more easily using standard methods of parameter estimation. One of the important questions we would like to

answer is the existence of a DAG structure that contains all the independence relationship of a probability distribution. In the next section we attempt to answer this question by discussing faithfulness and Minimality.

2.1.3 Faithfulness and Minimality Condition

What we asserted in the previous section is that if a DAG satisfies the Markov condition then it could be used to reduce the factorization of the joint distribution over a set of random variables. But is satisfying the Markov condition enough to presume reducibility? A DAG can satisfy the Markov condition and yet not reflect all the conditional independence relationships true among the random variables. Consider the DAG in Figure 2-3, suppose for a distribution, P over {X,Y,Z} for which the DAG satisfies the Markov condition we have the relation $X \perp Y$. This relation clearly, does not violate the Markov condition since X has no parents, yet the DAG does not reflect this constraint. The Markov condition though sufficient in reducing the factorization of the joint distribution may not entail all its dependencies.

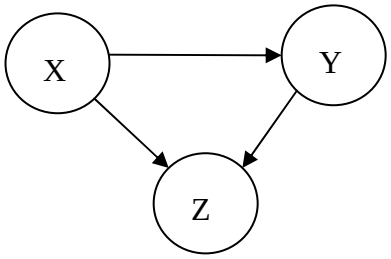


Figure 2-3: Illustrating Minimality condition

A stronger requirement is the Minimality condition which may be defined as follows:

> *Definition: (Minimality Condition)*
>
> *Let G, be a DAG with vertex set V and P a probability distribution on V generated by G. Then <G, P> satisfies the Minimality condition if and only if every proper sub graph H of G with vertex set V, the pair <H,P> does not satisfy the Markov condition.*

In other words, a DAG satisfies the Minimality condition if and only if it fails to satisfy the Markov condition by removing an edge.

The Markov condition applied to a graph produces a set of independence relations that usually entail other dependencies. A probability distribution over a set of random variables may also have some dependencies that are not entailed in applying the Markov condition to a DAG. If however, all and only the conditional independence relations that are true in the probability distribution, P are entailed in applying the Markov condition to

a DAG, G then the graph and the probability distribution are said to be faithful to each other. A formal definition is as follows:

> *Definition (Faithfulness)*
>
> *Let G be a DAG and P, a probability distribution generated by G. then <G, P> satisfies the faithfulness condition if and only if every conditional independence relation true in P, is entailed by the Markov condition applied to G.*

If a DAG satisfies the Markov and faithfulness condition then it implies the Minimality condition is satisfied. The Markov and Minimality condition do not however imply faithfulness. Ideally our goal would have been to learn faithful DAG's but these are not always easy to find. So for the purpose of simulating clinical avatars we will be comfortable with DAG's satisfying the Markov and Minimality condition.

2.1.4 D Separation

The Markov condition tells us the conditional independence relations necessary for reducing the factorization of the joint distribution of a set of random variables. However we may be interested in testing other dependencies which may not be obvious from a direct application of the Markov condition. For example consider the DAG in Figure 2-4, from the Markov condition we can detect the following independence relations,

$$
\begin{aligned}
&X \perp W \\
&Y \perp (H, Z, W) \mid X \\
&H \perp (Y, W) \mid X, Z \\
&Z \perp (Y, H) \mid X, W \\
&W \perp X, Y
\end{aligned}
\tag{2.1.5}
$$

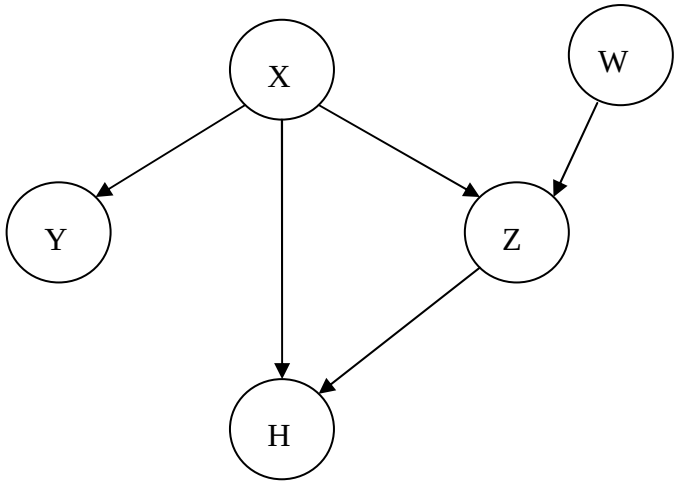


Figure 2-4: Illustrating D-Separation

Suppose we wanted to test the following independence relations which are not a direct consequence of the Markov condition,

$$
\begin{aligned}
&Y \perp W \\
&Y \perp W \mid Z \\
&Y \perp W \mid H
\end{aligned}
\tag{2.1.6}
$$

It is not clear how to arrive at a conclusion. The d-separation criteria help us to draw such conclusions. The relations $Y \perp W$ is really asking if the path from Y to W is blocked

without conditioning on any node. We will now discuss how paths between nodes may be blocked after which a formal definition of D-separation will be given. An observed node is one that has been conditioned upon and an unobserved node is without conditioning. If a path between two nodes is not blocked we will refer to it as active.

In the language of information theory, a path between two nodes is said to be blocked if information cannot flow from one node to another. Figure 2-5 presents a summary of different paths and the conditions under which they may be considered blocked.

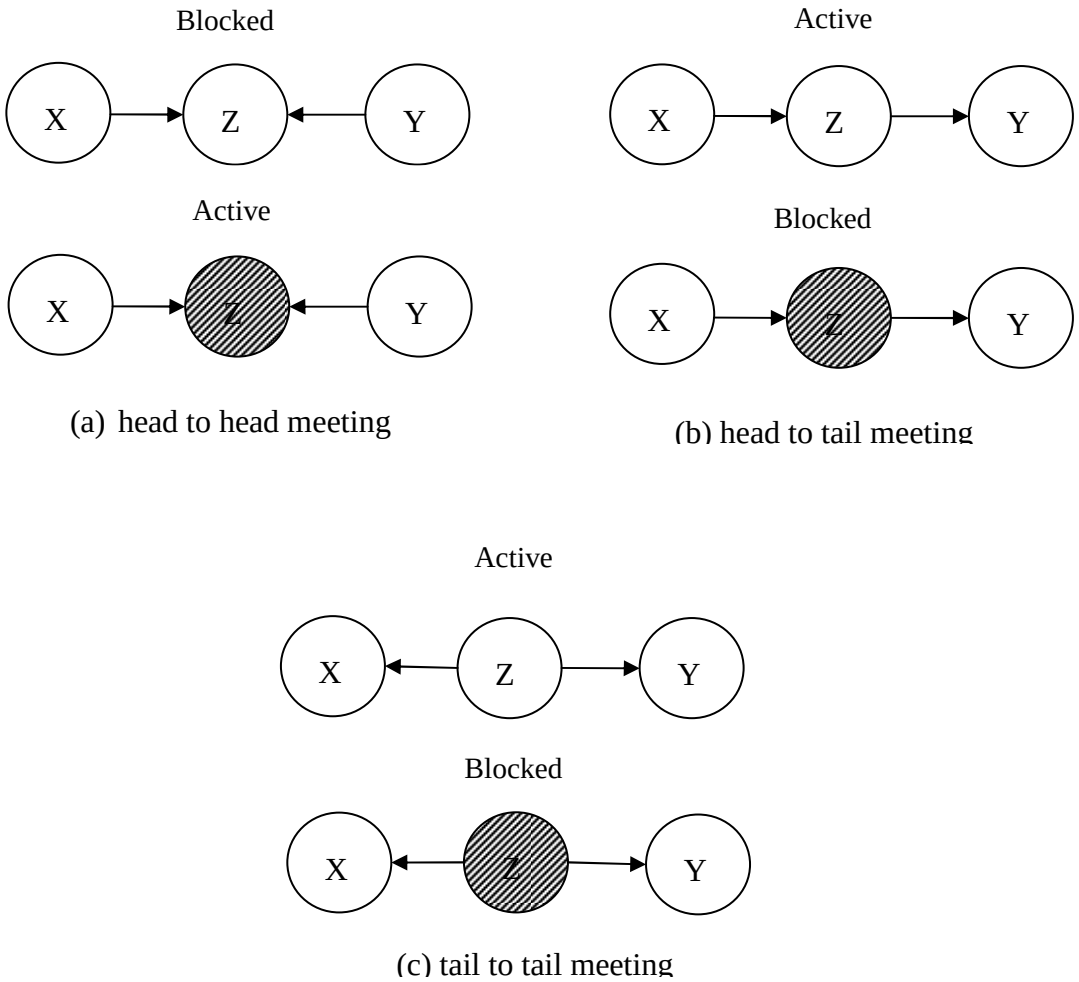


Figure 2-5: Blocked and Active paths illustrating d-separation

The path between X and Y illustrating a head to head meeting at *Z* Figure 2-5a is blocked when *Z* is unobserved but active when *Z* is observed. The head to tail meeting at *Z* illustrated in Figure 2-5b is active when *Z* is unobserved and becomes blocked when *Z* is

observed. Similarly, the tail to tail meeting in Figure 2-5c is active when Z is not observed and is blocked when Z is observed.

Suppose there is more than one node between X and Y as illustrated in Figure 2-6 then the path is blocked if any of the intermediary nodes renders it blocked.

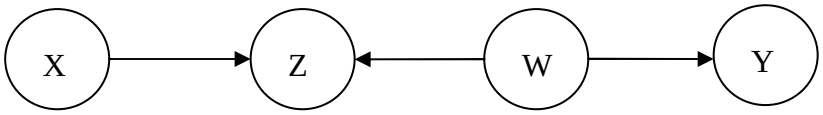


Figure 2-6: Illustrating d--separation by more than one node

Definition: (D-separation)

Two nodes X and Y in a directed acyclic graph are said to be d-separated by a non intersecting set of nodes C if all paths between X and Y are blocked when the nodes in C are observed.

Now returning to the independence relations in (2.1.6) we draw the following conclusions:

1) $Y \perp W$: The paths between Y and W are $< Y, X, Z, W >$ and $< Y, X, H, Z, W >$ which are all blocked when either H or Z are unobserved. Hence Y and W are d-separated without conditioning on any other node and the assertion holds

2) $Y \perp W|Z$: Conditioning on Z activates the path $< Y, X, Z, W >$ hence Y and W are not d-separated given Z and the conditional independence assertion fails to hold.

3) $Y \perp W|H$: Though conditioning on H activates the path $< Y, X, H, Z >$ Z still blocks the path $< Y, X, H, Z, W >$ and $< Y, X, Z, W >$ hence Y and W are d-separated given H and the conditional independence assertion holds.

2.2 Learning Bayesian Networks

So far we have assumed that we had a Bayesian network from which we estimated the joint distribution of the random variables. In this section we describe how Bayesian networks can be constructed. Specifying a Bayesian network involves:

1) Constructing the Directed Acyclic Graph
2) Estimating the parameters of the network

DAGs may be constructed directly from knowledge about the causal relationships between the random variables. These DAG's are commonly known as causal graphs(Pearl 2000). For example, consider a house fitted with an alarm system which goes off if either a burglar breaks into the house or there is an earthquake. There is a dog in the house which barks either when the Alarm goes off or it has fever. Let the random variable of interest be A-Alarm, B-Burglar, E-earthquake, D-Dog, F-Fever. To construct

a causal graph we will work our way down from causes to effects. The resulting causal structure is illustrated in Figure 2-7

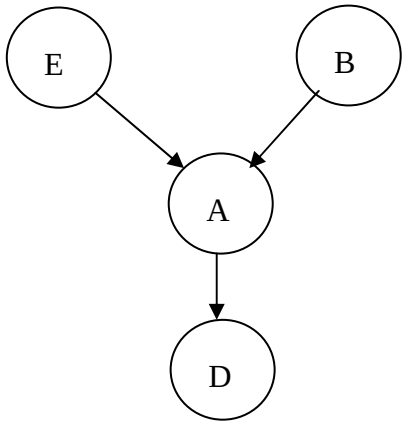


Figure 2-7: A Bayesian network constructed from causal knowledge

The second approach which is the direction of this work is in learning the structure.There are two major approaches to learning Bayesian networks from data, the Bayesian approach and the constrained based approach. In the Bayesian approach a score is assigned to DAG's in the space of possible Bayesian networks and the DAG with the highest score is returned. Constrained based algorithms on the other hand perform tests of conditional independence on all possible pairs of variables conditioned on every relevant subset of nodes, returning a structure which represents the independence relations that are true among the variables in the data set. We present a more detailed description of these methods in the next sections.

2.2.1 Bayesian Learning

Given a set of random variables $\{X_1, X_2, X_3 \cdots X_m\} = U$ and a dataset of examples of these variables$\{C_1, C_2, C_3 \cdots C_n\} = D$, suppose we wanted to determine$P(C|D, \xi)$, which is the probability distribution of a new case C, given the database D and our current state of information ξ. Assume also that the data D is a random sample from a distribution P, specified by an unknown Bayesian network structure, B_s. Let B_s^h denote the hypothesis that the data is generated by network structure B_s and that the hypotheses corresponding to all possible network structures form a mutually exclusive and collectively exhaustive set, then by laws of probability,

$$P(C \mid D, \xi) = \sum_{B_s^h} P(C, B_s^h \mid D, \xi) \tag{2.2.1}$$

From Bayes rule,

$$P(C, B_s^h \mid D, \xi) = \frac{P(C \mid B_s^h, D, \xi) P(B_s^h, D, \xi)}{P(D, \xi)} \tag{2.2.2}$$

Expanding the RHS further by obtain,

$$P(C, B_s^h \mid D, \xi) = P(C \mid B_s^h, D, \xi) P(B_s^h \mid D, \xi) \tag{2.2.3}$$

Substituting (2.2.3) into (2.2.1) we have,

$$P(C \mid D, \xi) = \sum_{B_s^h} P(C \mid B_s^h, D, \xi) P(B_s^h \mid D, \xi) \tag{2.2.4}$$

Obviously summing over all possible network structures may computationally impractical, hence we identify a subspace H containing Bayesian networks that account

for a high proportion of the hypotheses then posterior probability $P(C|D,\xi)$ can be approximated by,

$$P(C \mid D,\xi) \approx c \sum_{B_s^h \in H} P(C \mid D, B_s^h, \xi).P(B_s^h \mid D,\xi) \tag{2.2.5}$$

Where, c is a normalizing constant defined by,

$$c = \frac{1}{\sum_{B_s^h \in H} P(B_s^h \mid D,\xi)} \tag{2.2.6}$$

Clearly, $P(C|D,\xi)$ largely depends on the relative posterior probability$P(B_s^h|D,\xi)$. Hence the Bayesian learning task is to identify the subset H of network structures with a high posterior probability. When $|H| = 1$ we learn a single network structure, and a collection for $|H| > 1$. Equivalently we could search for the network structure with a high joint probability with the data set defined by,

$$P(D, B_s^h \mid \xi) = P(B_s^h \mid \xi)P(D \mid B_s^h, \xi) \tag{2.2.7}$$

Any formula which computes the relative posterior probability of a network-structure hypothesis is a Bayesian scoring metric which is discussed in more detail in section (2.3). Bayesian learning algorithms therefore comprise mainly of a scoring criterion which measures the relative posterior probability of a network hypothesis and search procedure for identifying such network structures.

In order the move sequentially through the search space the space must be divided into states. Each of the states will be represented by a DAG. The algorithms transition from

one state to another by removing an edge, adding an edge or reversing an edge. These edges are all directed edges. All operators are subject to the constraint that a cycle cannot be formed. Figure 2-8 illustrates how a search algorithm will move from one state to another using the operators mentioned.

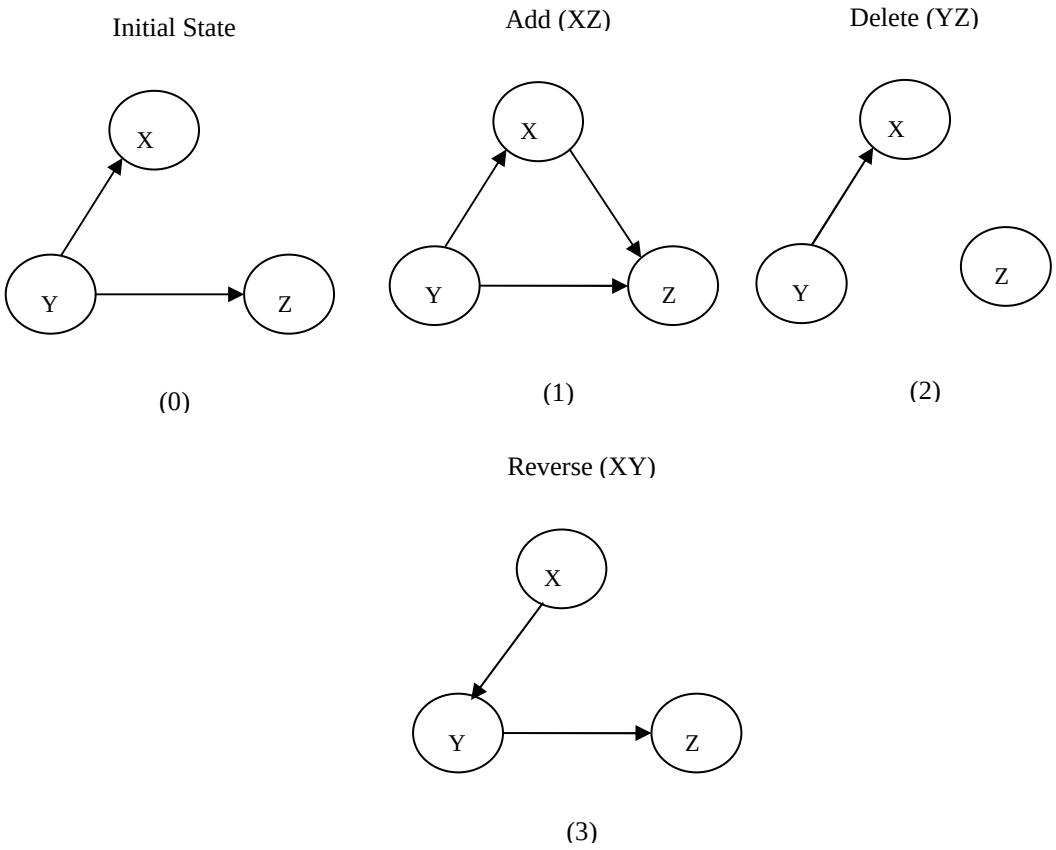


Figure 2-8: Search States of a Bayesian Learning Algorithm

At state (0) the algorithm performs any of the operations in (1), (2), (3) but only transitions if the score of the DAG resulting from the operation is higher than the score of the initial state. By sequentially applying (1), (2), (3) the optimum structure is identified.

For a graph with many nodes the task of traversing the B-space becomes quickly computationally expensive. To reduce this task, current search algorithms search through the space of equivalence classes (E-Space) where each state is a representation of an equivalence class of Bayesian networks and not an individual DAG. The operators for traversing this space are different from those used in the B-Space. Details of this approach can be found in (Chickering 2002).

The next important aspect of Bayesian learning is a scoring criterion by which each state will be evaluated. A scoring criterion takes as input a Bayesian network structure, a data set, and possibly some domain knowledge and returns a value indicating how well the structure fits the data. The more common scoring criteria interpret the Bayesian network as a set of assertions about the independence constraints that hold among a set of random variables. Such scoring criteria assign the same score to DAG's in the same equivalence class a property known as score equivalence. An important property scoring criteria must possess to efficiently identify an optimum DAG in the search space is decomposability.

Definition:

A Bayesian network structure scoring criterion is decomposable if it can be written as a sum of measures, each of which is a function only of one node and its parents. i.e.

$$S(G) = \sum_{i=1}^{n} s(x_i, \pi_{xi}) \quad (2.2.8)$$

Where π_{xi} , represents parents of node x_i. The property of being decomposable extremely simplifies the task of scoring each state in the search space. Instead of calculating the score of the entire DAG, decomposable scoring criteria would only need to score the nodes whose parents have changed as a result of the application of any of operations described. The more common scoring criteria used in the literature are, Bayesian information criteria, MDL criterion, AIC criterion, BDe criterion.

Another property of scoring criteria is score equivalence. We say a scoring criterion is score equivalent if it assigns the same score to DAG's in the same equivalence class. Since DAG's in a particular equivalence class have the same assertion of independence constraints, it makes sense for scores based on independence interpretation of structures to be score equivalent. Score equivalent criteria are thus sufficient for identifying a DAG that correctly estimates the joint distribution of the random variables. When the learning task is about identifying a causal structure we need more than score equivalent criteria. Score equivalent criteria are not able to distinguish between different members of the same equivalence class. Because an equivalent class can contain a wide variety of DAG's it is not sufficient to use score equivalent criteria when learning the causal network for a set of random variables More sensitive criteria have been developed that address this short fall and are able to distinguish DAG's in the same equivalent class. They are sensitive to direction of edges in the same equivalence class.

2.2.2 Constraint Based Learning

In constrained based learning, the structure of the Bayesian network is obtained by first performing test of conditional independence on different pairs of random variables to construct the skeleton (undirected graph) of the DAG. The edges in the skeleton are then oriented using a set of rules established by Christopher Meek. In this section we will provide a brief description of the construction of the skeleton and a summary of Meeks orientation rules.

Consider the joint space of random variables $U = \{X_1, X_2, X_3, X_4, X_5\}$ and the database of cases $D = \{C_1, C_2, \cdots, C_m\}$. Assume that the database was generated by the Bayesian network structure illustrated in Figure 2-9. The learning begins with the assumption that all the variables are dependent on each other, which is represented graphically by a complete undirected graph illustrated in Figure 2-10

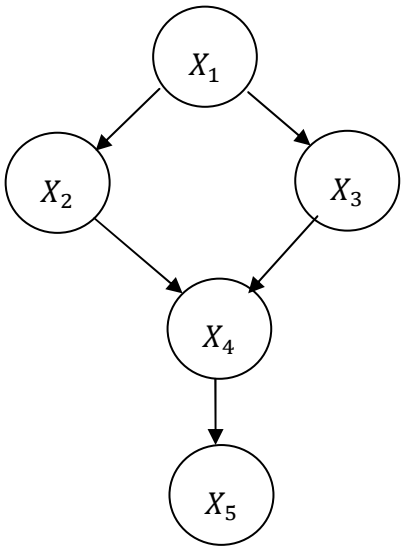


Figure 2-9: Gold Standard Bayesian Network

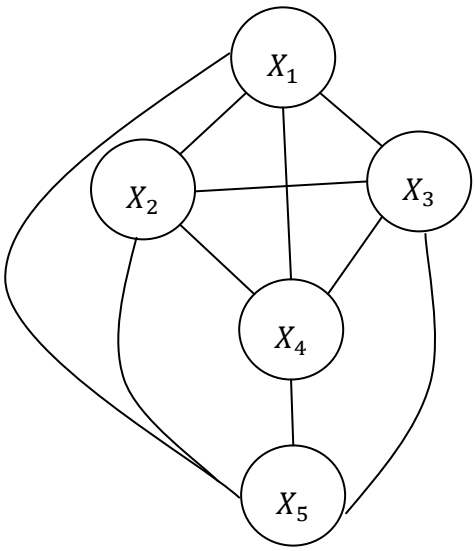


Figure 2-10: Complete Undirected Graph

Tests of conditional independence are then performed between pairs of variables. Initially, tests are performed directly without conditioning on any other variables. If any of the paired tests determine that two variables are independent the edge between them is removed. The next round of paired tests involves conditioning on a third node (variable). Suppose the test identified that $X_i \perp X_j | X_k$ then X_k is said to separate X_i and X_j and is stored in Sepset $(X_i, X_j) = \{X_k\}$ and the edge between X_i and X_j removed. Subsequent tests are performed by conditioning on larger sets until the size of the conditioning set exceeds the number of variables. At this point the first phase is complete and the skeleton is returned as illustrated in Figure 2-11

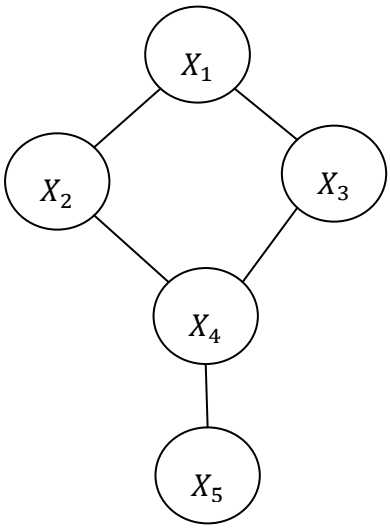


Figure 2-11: Skeleton of Gold Standard Network

With the complete separator set for each edge removed,

$$\begin{aligned} &Sep(X_1, X_5) = \{X_4\} \\ &Sep(X_2, X_5) = \{X_4\} \\ &Sep(X_3, X_5) = \{X_4\} \\ &Sep(X_1, X_4) = \{X_2, X_3\} \\ &Sep(X_2, X_3) = \{X_1\} \end{aligned} \tag{2.2.9}$$

Once the skeleton is obtained, the orientation phase begins by converting all triples to *unshielded colliders* where appropriate and following up with Meek's orientation rules. Unshielded colliders are commonly known as head to head meetings in the artificial intelligence literature$(X \rightarrow Y \leftarrow Z)$. To orient these, the algorithm, indentifies all unshielded triples of the form $(X - Y - Z)$. If y is not in the $sep(X, Z)$ then an arrow heads are drawn to$Y$, otherwise they are not oriented. Once all colliders are oriented the rest of the orientation is done to avoid the creation of more colliders and cycles. Figure 2-12 summarizes Meek's orientation rules. Orientation of colliders and the final orientation are illustrated in Figure 2-13. In Figure 2-13b the edges $X_3 \rightarrow X_1$ seems to have been reversed compared with the Gold standard. This very typical of constrained based learning because the orientation of colliders the rest of the orientation allows for a number of possible orientations. The theory suggests that any of the possible structures should be able to sufficiently generate the data. Standard statistical techniques are usually used in performing conditional independence tests. Measure such as mutual information

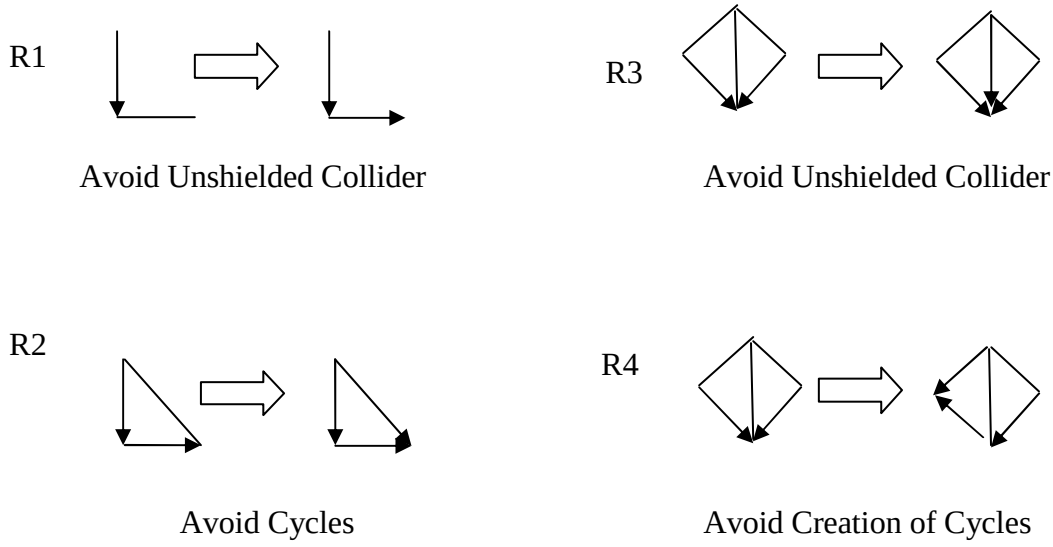


Figure 2-12: Meeks Orientation Rules

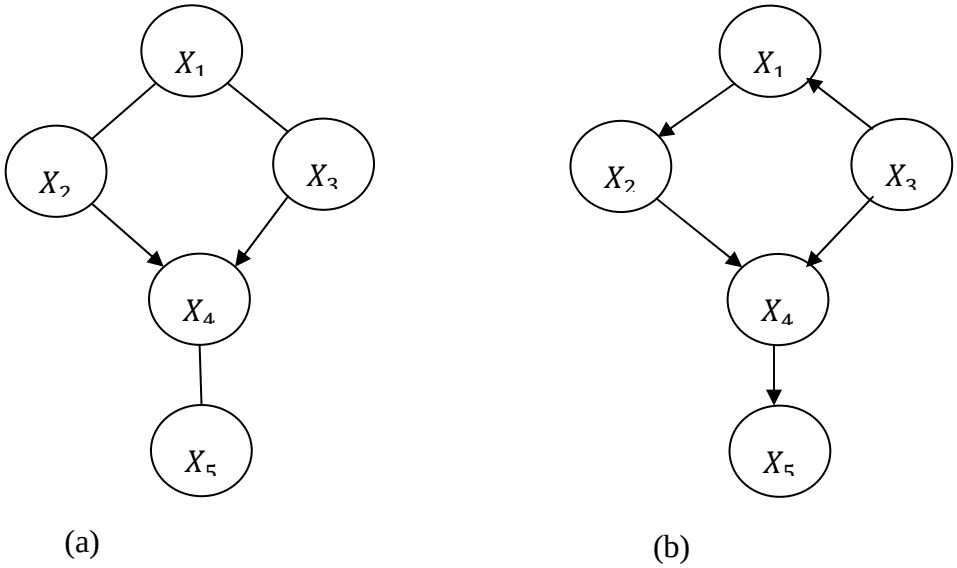


Figure 2-13: Orienting DAG's using Meeks rules, (a) Orienting colliders;

(b) Applying Meeks rules

2.3 Model selection

In this section we present a derivation of the Bayesian scoring metric and the underlying assumptions that guide its use. Some other useful metrics for selecting high scoring Bayesian networks for density estimation are also discussed.

2.3.1 Bayesian Scoring Metric (BDe)

Given the domain U of random variables and database D of cases, the Bayesian Scoring Metric as developed by (David Heckerman 1995) is a measure of the probability that a given dataset D was generated by the Bayesian network hypothesis B_s^h defined by,

$$P(D, B_s^h \mid \xi) = P(B_s^h \mid \xi)P(D \mid B_s^h, \xi) \tag{2.3.1}$$

Where $P(B_s^h|\xi)$ is the prior probability of the network hypothesis and $P(D|B_s^h, \xi)$ is the likelihood of the dataset given the network hypothesis.

Let,

D_l denote the first $(l-1)$cases in the database

r_i, be the number of states of the variable x_i

$q_i = \prod_{x_j \in \Pi_i} r_j$ be the total states of the parent set of x_i

$P(x_i = k \mid \Pi_i = j, \xi)$, the probability that $x_i = k$ given the j^{th}state of the parents of x_i

Set,

$$\theta_{i,j,k} = P(x_i = k|\Pi_i = j, \xi)$$

$\theta_{ij} = \cup_{k=1}^{r_i}\{\theta_{i,j,k}\}$, the parameter set for x_i over all its parents

$\theta_i = \cup_{j=1}^{q_i}\{\theta_{i,j}\}$, the parameter set for $x_i$over all states of its parent set

$\Theta_{BS} = \cup_{i=1}^{n}\theta_i$, the parameter set over all variables.

Consider also the following assumptions about the dataset D, and the network structure, B_s

1. (Multinomial Sample) For all network structures B_s in U there exists positive parameters Θ_{BS} such that, for $i = 1\cdots, n$ and for $k = k_1, \cdots k_{i-1}$

$$P(x_{il} = k \mid x_{1l} = k_1, \mathrm{L}\ \ x_{(i-1)l} = k_{i-1}, D_l, \Theta_{BS}, B_s^h, \xi) = \theta_{i,j,k} \qquad (2.3.2)$$

2. (Parameter Independence) Given a network structure B_s if $P(B_s^h|\xi) > 0$ then,

a. $\rho(\Theta_{B_s}|B_s^h, \xi) = \prod_{i=1}^{n}\rho(\Theta_i|B_s^h, \xi)$

b. For $i = 1, \cdots, n$: $\rho(\Theta_i|B_s^h, \xi) = \prod_{j=1}^{q_i}\rho(\Theta_{ij}|B_s^h, \xi)$

i.e. the parameters associated with variable in a network structure are independent as well as those associated with each parent.

3. (Parameter Modularity) Given two network structures B_{s1} and B_{s2} such that $P(B_{s1}^h|\xi) > 0$ and $P(B_{s2}^h|\xi) > 0$, if x_i has the same parents in B_{s1} and B_{s2}then,

$$\rho(\Theta_{ij} \mid B_{s1}^h, \xi) = \rho(\Theta_{ij} \mid B_{s2}{}^h, \xi) \quad j = 1, \cdots q_i \qquad (2.3.3)$$

i.e. the parameters Θ_{ij}depend only on the structure of the network that is local to the variable x_i

4. (Dirichlet Assumption) Given a network a structure $B_s$such that $P(B_s^h|\xi) > 0$, $\rho\left(\Theta_{ij}\middle|B_s^h,\xi\right)$ Dirichlet for all $\Theta_{ij} \subseteq \Theta_{Bs}$. That is there exists exponents $N'_{i,j,k}$, which depend on B_s^h and ξ, that satisfy

$$\rho(\Theta_{ij} \mid B_s^h, \xi) = c.\prod_k \theta_{ijk}^{N'_{ijk}-1} \tag{2.3.4}$$

Where c is a normalizing constant.

By the multinomial sample assumption and the assumption of no missing data, we obtain,

$$P(C_l \mid D_l, \Theta, B_s^h, \xi) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{ri}\theta_{ijk}^{1_{ijk}} \tag{2.3.5}$$

Extending this to the entire dataset and letting N_{ijk}denote the number of cases in database D such that $x_i = k$ and $\pi_i = j$ we have,

$$P(D_l \mid \Theta_{B_s}, B_s^h, \xi) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{ri}\theta_{ijk}^{N_{ijk}} \tag{2.3.6}$$

Hence by parameter independence the posterior distribution over the parameters of the network hypothesis can be estimated by,

$$\rho(\Theta_{B_S} \mid D, B_s^h, \xi) = c{\cdot}P(D \mid \Theta_{B_S}, B_s^h, \xi)\prod_{i=1}^{n}\prod_{j=1}^{q_i}\rho(\theta_{ij} \mid B_s^h, \xi) \tag{2.3.7}$$

Where c is some normalizing constant. Combining (2.3.7) and (2.3.6) we have,

$$\rho(\Theta_{B_S} \mid D, B_s^h, \xi) = c \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left[\rho(\theta_{ij} \mid B_s^h, \xi) \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}} \right] \tag{2.3.8}$$

By the assumption of i.i.d sample we have,

$$P(D \mid B_s^h, \xi) = \prod_{l=1}^{m} P(C_l \mid D_l, B_s^h, \xi) \tag{2.3.9}$$

Conditioning on the parameters of the network structure B_s we obtain,

$$P(C_l \mid D_l, B_s^h, \xi) = \int P(C_l \mid D_l, \Theta_{B_s} B_s^h, \xi) \cdot \rho(\Theta_{B_s} \mid B_s^h, \xi) d\Theta_{B_s} \tag{2.3.10}$$

Substituting (2.3.5) and (2.3.8)

$$P(C_l \mid D_l, B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \int \prod_{k=1}^{ri} \theta_{ijk}^{1_{ijk}} \left[\rho(\theta_{ij} \mid D_l, B_s^h, \xi) \right] d\theta_{ij} \tag{2.3.11}$$

When $1_{ijk} = 1$ the integral in (2.3.11) is the expected value of θ_{ijk}. consequently we have,

$$P(C_l \mid D_l, B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{ri} \left[E(\theta_{ij} \mid D_l, B_s^h, \xi) \right]^{1_{ijk}} \tag{2.3.12}$$

Substituting (2.3.12) into (2.3.9) we have

$$P(D \mid B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{ri} \prod_{l=1}^{m} \left[E(\theta_{ij} \mid C_1, \mathrm{L}\ C_{l-1}, B_s^h, \xi) \right]^{1_{ijk}} \tag{2.3.13}$$

By the Dirichlet assumption,

$$\rho(\theta_{ij} \mid D, B_s^h, \xi) = c.\prod_{k=1}^{r_i} \theta_{ijk}^{N'_{ijk}+N_{ijk}-1} \tag{2.3.14}$$

Where c is a normalizing constant. N_{ijk} are a sufficient statistic for the database. The posterior distribution of each parameter θ_{ij} remains in the Dirichlet family. Setting $l = m+1, c_{m+1} = C \; and \; D_{m+1} = D$ we obtain,

$$P(C_{m+1} \mid D, B_s^h, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \left(\frac{N'_{ijk} + N_{ijk}}{N'_{ij} + N_{ij}} \right)^{1_{m+1,ijk}} \tag{2.3.15}$$

Where,

$$N'_{ij} \equiv \sum_{k=1}^{r_i} N'_{ijk} \qquad N_{ij} \equiv \sum_{k=1}^{r_i} N_{ijk}$$

Finally we obtain the Bayesian Scoring Metric,

$$P(D, B_s^h \mid \xi) = P(B_s^h \mid \xi) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \tag{2.3.16}$$

2.3.2 Kullback-Leibler Distance (KLD)

Let $P(x_1, x_2, \cdots x_n)$ denote the joint distribution of the gold-standard domain and $q(x_1, x_2, \cdots x_n)$ denote the joint distribution of the next case to be seen as predicted by the learned networks$(i.e. P(C|D, \xi))$. The cross entropy $H(p, q)$ referred to as the Kullback-Leibler distance is given by

$$H(p,q) = \sum_{x_1 \mathrm{L}\, x_n} P(x_i, \mathrm{L}\ , x_n) \log \frac{P(x_i, \mathrm{L}\ , x_n)}{q(x_i, \mathrm{L}\ , x_n)} \tag{2.3.17}$$

Low values of KL-distance typically correspond to a learned distribution that is close to the gold standard. Its discrete form can be computed using the following relation,

$$H(p,q) = \sum_{i=1}^{q_i} \sum_{j=1}^{r_i} \sum_{k=1} P(X_i = k, \pi_i = j) \log \frac{P(X_i = k \mid \pi_i = j)}{q(X_i = k \mid \pi_i = j)} \tag{2.3.18}$$

The cross entropy measure reflects the degree to which the learned networks accurately predict the next unseen example in the data set or in other words how well it copies the true distribution. In chapter 3, we present a slight modification of this measure to facilitate model selection in our proposed methodology.

2.3.3 Mutual information

Mutual information is defined as a measure of the relationship between two random variables that are sampled simultaneously. You can also think of it as a measure of how much one random variable can tell you about another. The mutual information between two random variables is 0 if and only if they are independent i.e. they share no information. Consider two discrete random variables X and Y defined jointly by the distribution $P(X,Y)$, then the mutual information can expressed by the relation,

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log \frac{P(x,y)}{P(x)P(y)} \tag{2.3.19}$$

Where $P(x)$ and $P(y)$ represent the marginal distribution of the two random variables. By measuring the mutual information between each pair of variables in the gold standard network we able to compare this to a similar measure obtained using the learned networks and select a network that best preserves the interaction between variables. More details of this implementation of the mutual information for model selection are presented in chapter 3.

2.3.4 Bayesian Information Criterion

The Bayesian information criterion(Schwarz 1978) is a measure of the likelihood of the training data set given the associated parameters of a network structure i.e. $P(D|B_S)$. It is estimated using the method of maximum likelihood estimation. BIC contains a penalty term that punishes complex models that may be over fits of the distribution of the dataset. It is defined according to(Mozaherul Hoque Abul Hasanat 2010) as,

$$Q_{BIC} = log(P(B_s)) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ij} log \frac{N_{ijk}}{N_{ij}} - Pen(N)Dim(B_s) \qquad (2.3.20)$$

Where, $Pen(N) = \frac{1}{2} log(N)$ and $Dim(B_s) = \sum_{i=1}^{n} q_i(r_i - 1)$ together represent the penalty term.

2.4 Parameter Estimation

Once the structure of the Bayesian network has been obtained and the relevant conditional independence relationships extracted by either the Markov condition or D-separation, it remain to estimate the respective conditional distributions. We will discuss two popular methods of estimating conditional distributions from data: maximum likelihood estimation and maximum a posterior estimation.

2.4.1 Maximum Likelihood (ML) Estimation

Consider a random variable X, distributed according to a known parametric distribution *Dist* with parameter $\boldsymbol{\mu}$. Let $D = \{x_1, x_2, \cdots x_n\}$ be a database of i.i.d cases of the random variable. Then the maximum likelihood estimate of the parameter $\boldsymbol{\mu}$ is the setting of $\boldsymbol{\mu}$ that maximizes the probability of the data set, often referred to as the likelihood function (L ($\boldsymbol{\mu}$)) which is expresses as,

$$L(\mu) = \Pi_{i=1}^{N} \mathrm{P}(x_i \mid \mu) \tag{2.4.1}$$

Suppose X is a Bernoulli random variable and μ defines the probability of a success. Then the likelihood function becomes,

$$L(\mu) = \Pi_{i=1}^{N} \mu^{x_i} (1-\mu)^{1-x_i} \tag{2.4.2}$$

It is usually much easier to maximize the log likelihood function which results in the same ML estimate by the monotonicity of the logarithm. It follows that,

$$\log L(\mu) = \sum_{i=1}^{n} x_i \ln \mu + (1 - x_i) \ln(1 - \mu) \tag{2.4.3}$$

Differentiating the RHS and setting the result equal to zero we have,

$$\sum_{i=1}^{n} \frac{x_i}{\mu} = \sum_{i=1}^{n} \frac{(1 - x_i)}{1 - \mu} \tag{2.4.4}$$

Solving for μ we have,

$$\sum_{i=1}^{n} x_i \left(\frac{1-u}{u} + 1 \right) = N \tag{2.4.5}$$

which implies,

$$\mu^{ML} = \frac{1}{N} \sum_{i=1}^{n} x_i = \frac{m}{N} \tag{2.4.6}$$

Where, m is the number of successes. Thus the maximum Likelihood estimate for the probability of a success of a Bernoulli random variable is the proportion of success. The maximum likelihood estimate is biased with insufficient data, however converges to the true distribution in the limit of large data

2.4.2 Maximum a posterior Estimation

When prior knowledge about the parameters of a conditional distribution is available, it is important to use these in estimating the true distribution. Maximum a posterior estimation seeks to maximize the posterior distribution over the parameters given data on a given set of random variables.

Again let X be a random variable and $D = \{x_1, x_2, \cdots x_n\}$ a dataset of cases, then the posterior distribution is given by,

$$P(\mu \mid D) = \frac{P(\mu).P(D \mid \mu)}{P(D)} \tag{2.4.7}$$

Where $P(\mu)$ denotes the prior distribution over the parameter μ and $P(D|\mu)$ the likelihood. Since $P(D)$is independent of μ we normally have the relation,

$$Posterior \propto Prior \times Likelihood$$

In order to simplify the estimation of the posterior distribution, we often choose priors that have a similar form as the likelihood function. These are usually referred to as conjugate priors. For example the Beta distribution is a conjugate prior the parameter of a Bernoulli random variable. Likewise, the Dirichlet distribution for the multinomial random variable. Suppose X is a Bernoulli random variable with probability of success μ, and $Beta(a_0, b_0)$ prior, then the posterior distribution over μ can be expressed as,

$$P(\mu \mid a_0, b_0, D) = cP(D \mid \mu)P(\mu \mid a_0, b_0)$$

$$= c\left(\Pi_{i=1}^{N} \mu^{x_i}(1-\mu)^{1-x_i}\right) Beta(\mu \mid a_o, b_o)$$

$$= c\frac{\Gamma(a_0+b_0)}{\Gamma(a_0)\Gamma(b_0)}\left(\mu^{\sum x_i + a_0 - 1}(1-\mu)^{\sum(1-x_i)+b_0-1}\right)$$

Which is a beta distribution with number of success $= m + a_0 - 1$

The maximum value of the posterior distribution is obtained at the maximum likelihood estimate of μ, which has been show to be $\frac{m}{N}$. Hence the Map estimate for μ would be,

$$\mu^{MAP} = \frac{m + a_0 - 1}{N + b_0 + a_0 - 2} \tag{2.4.8}$$

Chapter 3: Iterative Knowledge Guided Search

In this section we describe the main contribution of this work: The *Iterative Knowledge Guided Search* (IKGS).Though we are primarily interested in estimating the joint distribution of a given set of random variables, we would also like to extract as much causal knowledge *(in this application statistical dependencies)* from the dataset as possible to help increase our understanding of the domain. The performance of most search algorithms is largely domain dependant as explained in (Mozaherul Hoque Abul Hasanat 2010) making it difficult to identify a *'best algorithm'*. Algorithms that may correctly capture the joint distribution of the data set may not always present the true underlying causal network. The IKGS approach attempts to combine expert knowledge of a domain with the outputs of a collection of search algorithms to obtain a structure that accurately estimates the joint distribution as well as present us with substantial causal knowledge of the domain. We have explained this approach within the environment TETRAD(Clark Glymour), a software for constructing Bayesian networks. The method can however be easily implemented in other software packages.

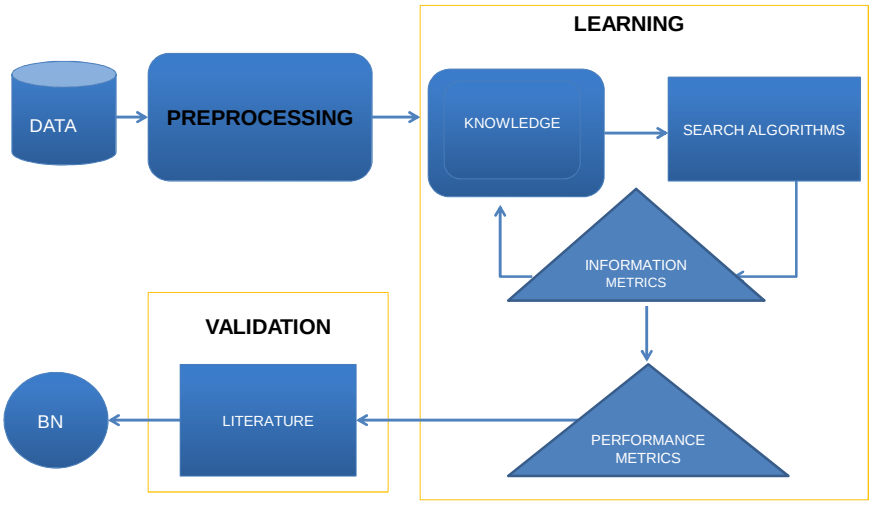


Figure 3-1: Workflow for Iterative Knowledge Guided Search

Figure 3-1 is a workflow that describes how using IKGS we can transition from data on a set of random variables to a Bayesian network model. There are three major stages in this process:

1. Preprocessing of Data
2. Learning
3. Validation

3.1 Preprocessing of Data

Before any attempt is made to learn a Bayesian network model it is important to improve the quality and workability of the data set. Search algorithms that learn Bayesian

networks essentially look for dependencies among the random variables making it imperative to ensure that any factors that might distort possible dependencies are eliminated. Common preprocessing tasks may include: data cleaning; dimension reduction and imputation of missing values. Missing values don't usually affect the effort to train a Bayesian network however if the values are not missing at random and account for more that 30% of the data it would be advisable to impute the missing values. This will ensure that the correct distribution of variables is used in training the Bayesian network. Other preprocessing efforts will depend on the data set and the intended use of the Bayesian network. 70% of the data is sufficient for training and 30% for testing. This is however subject to the size of data available.

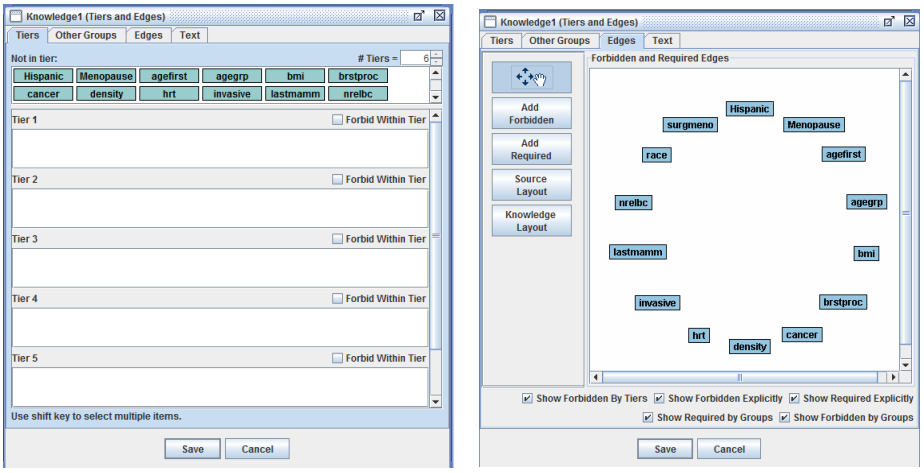
3.2 Learning

The learning phase is carried out in two stages. First, candidate Bayesian networks are trained using a collection of search algorithms and then we evaluate their performance in classifying unseen data (test data) to select a final model.

3.2.1 Training

During training, prior knowledge of the dependencies (edges) between the random variables are entered into TETRAD in the form of tiers and edges. The tiers define which random variable can potentially influence others while the edges enforce dependencies that must appear in or be absent from all learned networks. Figure 3-2 provides an

illustration of this concept in TETRAD. Six search algorithms, based on both score and constrained based approaches are used to train individual Bayesian networks. A summary of the algorithms used is illustrated in Table 3-1. Each candidate Bayesian network is scored using the Kullback- Leibler (KL) distance, Mutual Information (MI) and the Bayesian Information Criteria (BIC) as training metrics. These metrics evaluate how closely the Bayesian Network Model (BNM) approximates the true distribution. The models with the best scores are selected and their common edges are used to update the knowledge base. The process is repeated until no more common edges can be identified.



(a) Tiers (b) Edges

Figure 3-2: TETRAD knowledge box illustrating tiers and edges. Variables in upper tiers can influence variables in lower tiers

We use search algorithms with different heuristic approaches to ensure that edges in the final model are highly significant. By consistently updating our prior knowledge the search space is constrained and the algorithms are able to detect more significant edges.

SEARCH ALGORITHM	DESCRIPTION
PC (Peter Spirtes 2000) *(Peter Clark)*	Basic Constraint based algorithm
PCLINGAM	Takes the output of PC algorithm and the training data and attempts to improve orientation
CPC *(Conservative PC)*	Variant of PC algorithm that improves orientation
JPC (Ramsey 2010) *(Joseph's version of PC)*	Runs iterations on the output of PC until convergence
JCPC *(Joseph's Version of CPC)*	Same as JPC but with PC
GES(Chickering 2002) *(Greedy Equivalence Search)*	Score based Algorithm

Table 3-1 : Summary of Search Algorithms

3.2.2 Performance Evaluation

Once the training phase is over, the best resulting DAG's are used in turn to classify each variable in the test data set. The proportion of correct classification defined as the ratio of correctly classified cases to the total cases, is computed and averaged out across all variables. This gives a measure of how well each of the candidate Bayesian networks can predict unseen data. The results are interpreted as the higher the proportion of correct classification the better the network. By comparing these results to those obtained in the final stage of training we select a final Bayesian network.

3.3 Validation

The validation stage involves a comparison of the edges identified by our final model with published dependencies between the random variables of interest.

We initially classify edges as being

1. Validated by Literature
2. Rejected by Literature
3. Without Evidence

For edges without evidence we consult with domain experts to determine their significance. Edges in the literature which are undetected by our model may be added and the performance of the model re-evaluated. The validity of our model is then measured as a ratio of the total number of edges validated by literature to the total number of edges learned from the dataset.

Chapter 4: Application to Breast Cancer Risk Prediction

Our goal is to use data from the breast cancer surveillance consortium to develop a Bayesian network which would present the dependencies between breast cancer risk factors and from which we can simulate clinical avatars to interrogate already existing risk prediction algorithms.

4.1 Data Description and Preprocessing

Our data set originally contained 2,392,998 records of index screening mammograms from women included in the Breast Cancer Surveillance Consortium(Barlow WE 2006). There were a total of fourteen variables describing various pathological and mammography characteristics of the women. These variables have been determined to influence a woman's risk for developing breast cancer and will henceforth be referred as risk factors. The variables include information on the women who developed breast cancer after a one year follow up. An extra training variable was included to determine which of the record was suitable for training and which for validation. The size of the data set was reduced to 302,355 records by introducing a count variable indicating the frequency of each combination of patient characteristics.

The data set was reverted to its original size by using the count variable. A total of 150,000 records were sample from original data for training (90%) and validating (10%) the Bayesian network, henceforth referred to as sample data. The data set was stratified

using the cancer variable and a simple random sample was taken from each stratum in proportion to the original distribution of the cancer variable. Histogram plots were used to ensure that sampled data did not distort the original distribution of the data as illustrated in Figure 4-2. A total of 21.44% of the data was missing with 12,375 complete records. Table 3-1 provides a brief description of the variables in the data set and the number of records that were missing.

The training and count variables were removed from the data set after they had been used. Typically Bayesian networks can be trained using incomplete data (with missing values), however the distribution of missing values must be random and present no sample bias. To avoid errors that may result from data not being missing at random we imputed the missing values using multiple imputations. Figure 4-1 describes the multiple imputation process.

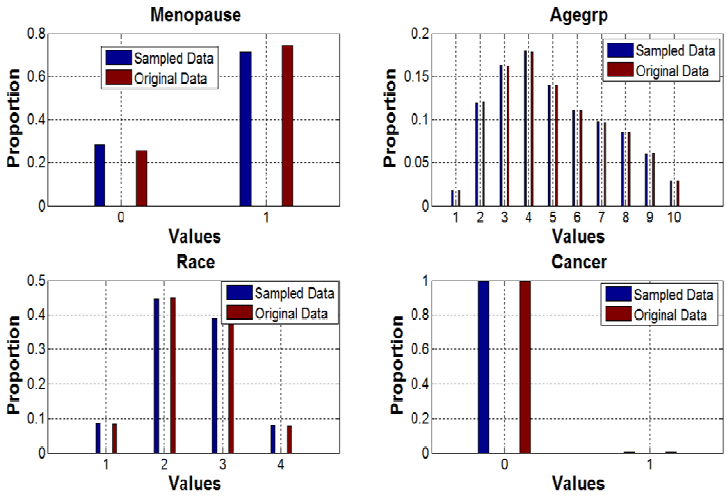


Figure 4-2: Comparison of sampled data with original data

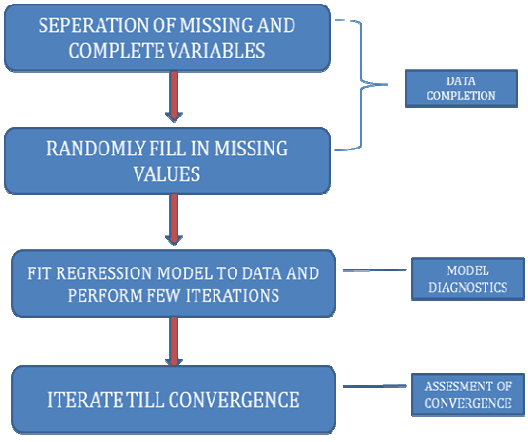


Figure 4-1: Multiple Imputations with Chained Equations

	Variables	Description	include	order	number.mis	all.mis	type	collinear
1	Menopause	Menopausal status	Yes	1	11397	No	binary	No
2	Agegrp	Age group	Yes	NA	0	No	positive-continuous	No
3	Density	Breast density	Yes	2	40755	No	ordered-categorical	No
4	Race	Race	Yes	3	23725	No	ordered-categorical	No
5	Hispanic	Hispanic	Yes	4	30554	No	binary	No
6	Bmi	Body Mass Index	Yes	5	83732	No	ordered-categorical	No
7	Agefirst	Age at first life birth	Yes	6	83408	No	ordered-categorical	No
8	Nrelbc	Number of relatives with first degree breast cancer	Yes	7	22863	No	ordered-categorical	No
9	Brstproc	Previous breast procedure	Yes	8	15431	No	binary	No
10	Lastmamm	Result of last mammogram before index mammogram	Yes	9	34983	No	binary	No
11	Surgmeno	Surgical menopause	Yes	10	78234	No	binary	No
12	Hrt	Current hormone therapy	Yes	11	61514	No	binary	No
13	Invasive	Diagnosis of Invasive Breast Cancer	Yes	NA	0	No	binary	No
14	Cancer	Diagnosis of invasive or ductal carcinoma in situ breast cancer within one year of index screening mammogram	Yes	NA	0	No	binary	No
15	Training	Training/Testing	Yes	NA	0	No	binary	No

Table4-1: Description of Variable

We ensured that the distribution of imputed variables was similar to that of the actual variables. Histograms of the distribution of imputed data against observed data illustrated in Figure 4-3 show that the imputed data preserved the original distribution. Three Imputed data sets resulted from the imputation process and one was selected at random for training the Bayesian networks.

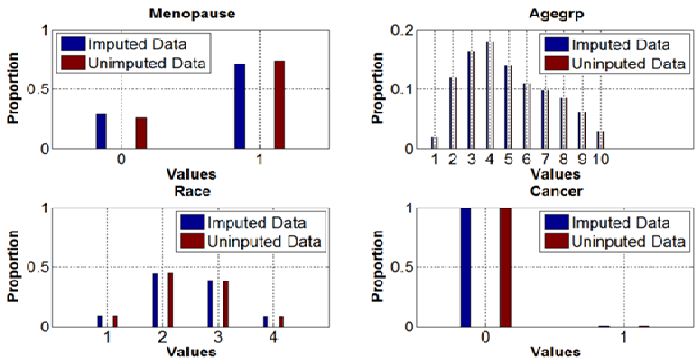


Figure 4-3: Comparison of sampled data with imputed data

4.2 Learning

4.2.1 Training

Six search algorithms as described in chapter 3 were collectively used in training six Bayesian network models (BNM). For each iteration, the models with the highest BIC score, Mutual Information and KL Distance were selected as candidate Bayesian networks. The common edges in these graphs were added to the prior knowledge. The cycle of search and knowledge update continued until no more common edges were detected. The model generated by each algorithm is denoted by NAME-BNM. For example the model generated by the GES algorithm will be denoted GES-BNM. The prior knowledge of the structure of the Bayesian network, used for the search, was obtained by interviewing experts in Breast Cancer research. This information was entered in TETRAD in the form of tiers and Edges as illustrated in Figure 4-4

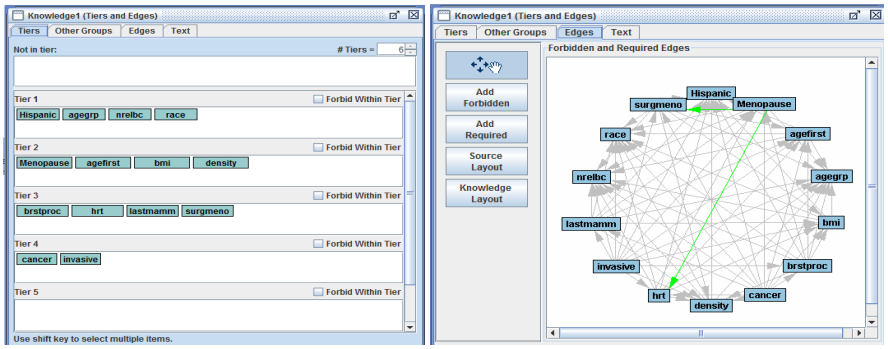


Figure 4-4: Entering Prior Knowledge into TETRAD

Out of the 135,000 (90%) of the sample data designated for training, 25,000 records were sampled successively to train the six Bayesian networks on each iteration. This was done primarily to reduce the learning time. A total of five iterations were performed after which no more common edges could be detected. The knowledge updates for each iteration are presented in Figure 4-5. Table 4-2 shows the results of metrics at each iteration.

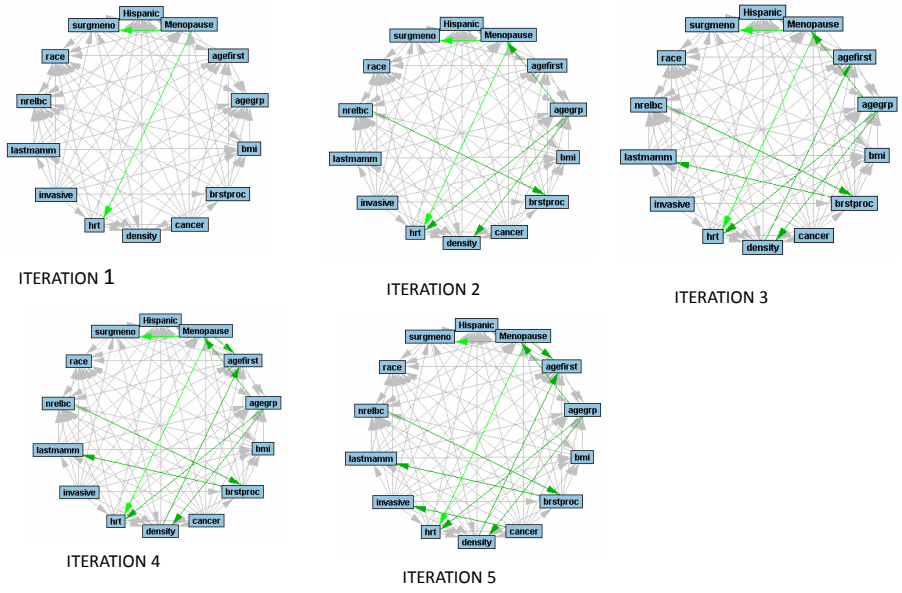


Figure 4-5: Knowledge updates per Iteration

SEARCH ALGORITHM	ITERATION 1				ITERATION 2				ITERATION 3			
	BIC	KL	MI	Edges	BIC	KL	MI	Edges	BIC	KL	MI	Edges
PC	-220,987.52	30.22	0.023	23	-223,597.62	31.52	0.074	24	-223,542.43	32.21	0.072	27
PCL	-222,424.35	27.28	0.023	23	-223,597.62	33.06	0.074	24	-226,289.56	30.41	0.012	27
CPC	-223,611.83	29.06	0.019	25	-233,609.41	31.71	0.040	24	-223,839.50	31.10	0.046	23
JPC	-219,784.52	28.31	0.035	24	-224,654.84	33.94	0.022	22	-225,538.82	36.82	0.074	27
JCPC	-220,824.12	30.74	0.046	24	-221,273.39	34.83	0.046	24	-224,150.76	36.30	0.026	25
GES	-217,873.28	33.13	0.035	12	-217,243.18	33.61	0.026	16	-217,617.16	34.10	0.026	15

	ITERATION 4				ITERATION 5			
	BIC	KL	MI	Edges	BIC	KL	MI	Edges
PC	-223,876.30	35.00	0.017	24	-223,398.03	31.80	0.076	24
PCL	-225,512.45	33.00	0.014	25	-225,814.30	33.00	0.079	26
CPC	-223,162.39	31.40	0.018	22	-223,452.35	31.50	0.016	22
JPC	-231,325.28	31.00	0.019	25	-224,882.59	33.50	0.075	25
JCPC	-221,202.44	31.50	0.017	24	-227,768.35	32.70	0.017	25
GES	-217,007.64	33.80	0.033	14	-217,174.09	36.80	0.031	16

Table 4-2 : Results of metrics for each iteration

At the end of the fifth iteration, the best performing models were CPC-BNM, JPC-BNM GES-BNM. GES-BNM had both the highest BIC score (-217,174.092) and KL distance (36.80). CPC-BNM had the highest mutual information (0.0164). We added JPC-BNM

which had the second highest KL distance (33.50) so we could have three models for performance evaluation. The DAG's of these three models are illustrated in Figure 4-6.

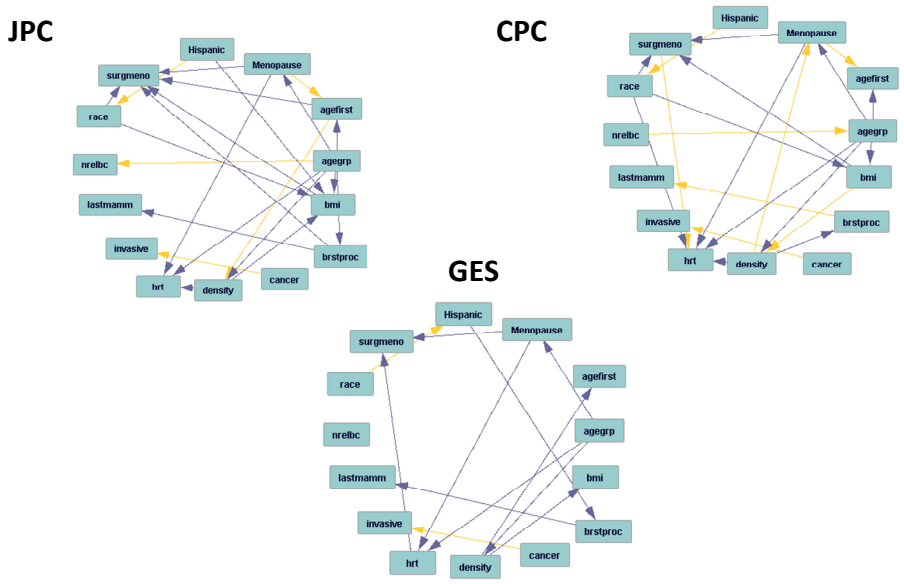


Figure 4-6: Best Performing DAG's

4.2.2 Performance Evaluation

Each of the candidate Bayesian networks obtained from the training phase was used in turn to classify all the variables in the data set. The classification rates for each model were obtained by averaging the rates across all variables. CPC-BNM correctly classified 75.65% of the data, while JPC-BNM and GES-BNM classified 75.27% and 74.99% respectively.

In selecting the final Bayesian network we were interested in a model which was parsimonious, had produced a relatively close estimation of the joint distribution of risk factors and performed relatively well in predicting unseen data. Although CPC-BNM had a relatively high classification rate (75.65%), its KL distance was relative low (31.50) and had a total of 22 edges. JPC-BNM and GES-BNM seemed to perform equally in classification (75.27%/74.99%), but GES-BNM's higher KL distance (36.80/33.50) and fewer edges (16/25) make it the more desirable candidate of the two. Since the primary purpose of the learned network is to generate clinical avatars consistent with the distribution of the dataset and not for classification, we choose GES-BNM as our final model.

4.3 Validation

In order to validate our final model, we mined the literature on associations between breast cancer risk factors and constructed a Bayesian network whose edges were based on our findings. Figure 4-7 shows our final model and the mined model.

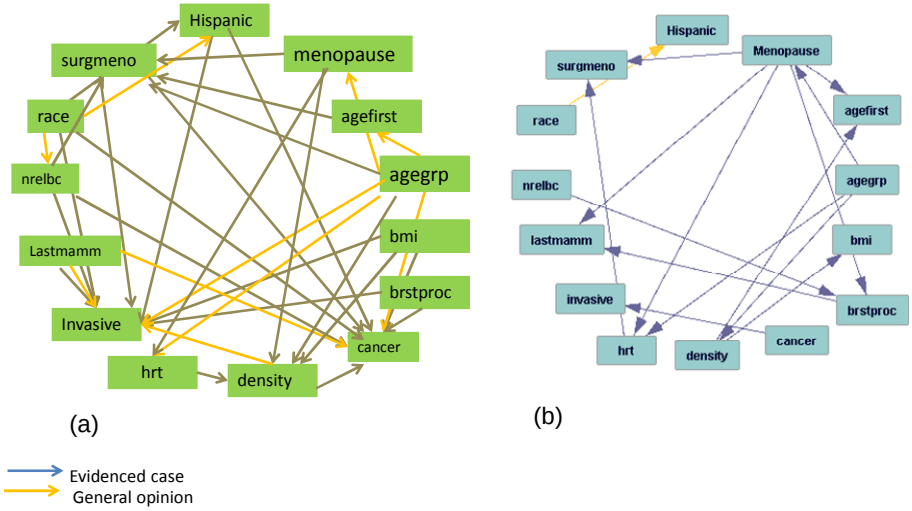


Figure 4-7: Mined Model (a) and IKGS model (b)

A total of nine risk factors each with directed edges to cancer and invasive, were absent from our model. We suspect that the very small proportion of women in our data set which developed cancer (0.04%) may have been insufficient to detect any reasonable correlation of cancer or invasive with the risk factors. We considered this a defect of our data set and not our learning approach. To create a more level playing field we removed all the edges to cancer and invasive in the mined network and compared the resulting graph with our final model. Figure 4-8 shows the reduced model.

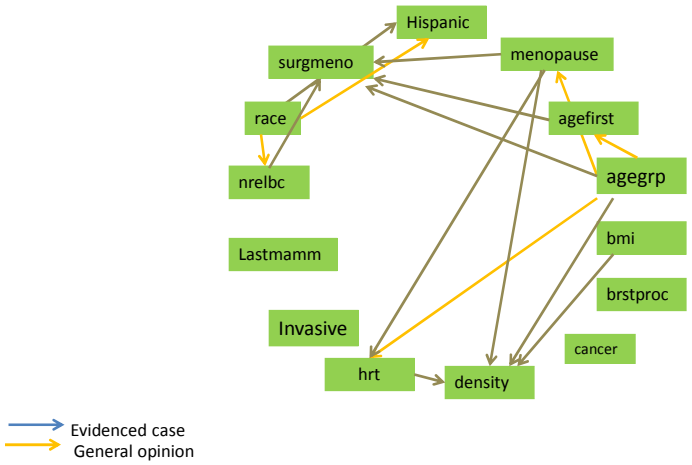


Figure 4-8: Reduced mined model

While the reduced mined model had 14 edges the IKGS model had 16 edges. Seven of mined dependencies were correctly detected by the IKGS model. The directed edge between bmi and density was reversed in the IKGS model. The remaining seven were not detected by the IKGS model. There were also seven new edges detected by the IKGS model not supported by literature. We propose these as potential dependencies that should be investigated by domain expects. To assess how close the IKGS model was to the mined model, we computed the Kl-distance and BIC scores using the mined model and compared with the results we had with the IKGS model. The results as shown in Table 4-3 illustrate that IKGS had higher likelihood of generating the training data than

the mined model (BIC:-220093.98/-216617.90). The distribution estimated by the mined model was however closer to the true distribution than the IKGS model (KL: 39.70/36.80).

Model	*BIC*	*KL*	*NUM EDGES*
Mined-BNM	-220,093.9879	39.74	14
IKGS-BNM	-216,617.90	36.80	16
PC-BNM	-220,987.5207	30.22335	23
PCL-BNM	-222,424.3548	27.28397	23
CPC-BNM	223,611.8346	29.06333	25
JPC-BNM	-219,784.521	28.31646	24
JCPC-BNM	-220,824.1186	30.73635	24
GES-BNM	217,243.1787	33.60531	12

Table 4-3: Comparing metrics of Mined model and IKGS model

Chapter 5: Conclusion and Recommendation

5.1 Conclusions

We have developed a search approach that harnesses the strengths of already existing algorithms to learn a Bayesian network that produces an improved estimation of the joint distribution of a set of random variables. Using the Bayesian information criteria, Kullback-Leibler distance and Mutual information, we have selected a model that closely matches the distribution of a dataset. By consistently updating our prior knowledge of the true structure of the Bayesian network we are able to produce a model whose edges are consistent with the independence relations that hold in the true distribution. We have applied this approach to learn the Bayesian network for breast cancer risk factors, which will be used in simulating clinical avatars (artificial patient populations) for interrogating various risk prediction models. We have shown using the Kullback-Leibler distance and Bayesian information Criterion that our final model learned with the Iterative Knowledge Guided Search (IKGS) is a better estimate of the distribution of risk factors compared with the output of any single search algorithm. By comparing our IKGS model with a mined model constructed from published breast cancer studies, we have shown that our model agrees with literature on breast cancer.

5.2 Recommendations

The Iterative Knowledge Guided Search (IKGS) approach to learning Bayesian networks is far from fully developed. For improved performance we have made the following recommendation:

1. A more precise implementation of the Kullback-Leibler distance should be used in model selection.
2. Search Algorithms that are likely to produce similar outputs should be removed from the algorithm set to reduce learning time.
3. The methodology should be extended to learn Bayesian networks in causally insufficient domains.
4. A more rigorous validation of the final learned model should be performed especially to assess the consistency of the learned dependencies with the true distribution. We suggest a direct checking of the independence relations obtained by d-separation with the conditional probability table. Chi-square tests could also be performed on the simulated avatars to check consistency.
5. We also believe that using all the records with women who developed breast cancer for training and testing the Bayesian network may provide a better reflection of the dependencies between the risk factors. Better still, the original data set could be sampled to support current statistics of 12% of women at risk of developing cancer.

6. IKGS is still quite manual and would be considerably more efficient if a single algorithm was written for the entire process.

Appendix

A1: Essential Algorithms

Pseudo code for PC algorithm (Constrained Based Algorithm)

A) Form the complete undirected graph C on the vertex set V.

B)

n=0,

repeat

repeat

select an ordered pair of variables X and Y that are adjacent in C such that Adjacencies(C,X)\{Y} has cardinality greater than or equal to n, and a subset S of Adjacencies(C,X)\{Y} of cardinality n, and if X and Y are d-separated given S delete X-Y from C and record S in Sepset(X,Y) and Sepset(Y,X) until all ordered pairs of adjacent variables X and Y such that Adjacencies(C,X)\{Y} has cardinality greater than or equal to n and all subsets S of Adjacencies(C,X)\{Y} of cardinality n have been tested for d-separation;

n = n+1;

until for each ordered pair of adjacent vertices X,Y Adjacencies(C,X)\{Y} is of cardinality less than n.

C) For each triple of vertices, X,Y,Z such that the pair X,Y and the pair Y,Z are each adjacent in C but the pair X,Z are not adjacent in C, orient X-Y-Z as $X \rightarrow Y \leftarrow Z$ if and only if Y is not in Sepset(X,Z)

D) Repeat

If $A \rightarrow B$, B and C are adjacent, A and C are not adjacent, and there is no arrow head at B, then orient B-C as $B \rightarrow C$.

If there is a directed path from A to B, and an edge between A and B, then orient A-B as $A \rightarrow B$.

Until no more edges can be oriented. (Peter Spirtes 2000)

Matlab codes for Data preprocessing

```
% Algorithm to replace missing values

for i = 1: size(data,2)
    if i == 2
        %do nothing
    else
        data(find(data(:,i)==11),i)= NaN;
    end
end
```

```
% Produces normalized histogram of between variables in two data sets

%generate proportions
n1=hist(data1(:,1));
n1 = n1(:,find(n1));
n2=hist(data2(:,1));
n2 = n2(:,find(n2));
y = unique(data1(:,1));
```

```
x =[(n1./sum(n1))' (n2./sum(n2))'];

  set(0,'defaultaxesfontsize',20);
  subplot(2,2,1); bar(y,x,.25,'hist'); % <- percentage cum dist
     ylabel('\bf\fontsize{20} Proportion');
     xlabel('\bf\fontsize{20} Values')
     title('\bf\fontsize{20} Menopause')
     legend('Sampled Data','Original Data')
     grid on

     %% variable 2
     n1=hist(data1(:,2));
n1 = n1(:,find(n1));
n2=hist(data2(:,2));
n2 = n2(:,find(n2));
y = unique(data1(:,2));
x =[(n1./sum(n1))' (n2./sum(n2))'];

  set(0,'defaultaxesfontsize',20);
  subplot(2,2,2); bar(y,x,.25,'hist'); % <- percentage cum dist
     ylabel('\bf\fontsize{20} Proportion');
     xlabel('\bf\fontsize{20} Values')
     title('\bf\fontsize{20} Agegrp')
     legend('Sampled Data','Original Data')
grid on

     %% Variable 3
     n1=hist(data1(:,3));
n1 = n1(:,find(n1));
n2=hist(data2(:,3));
n2 = n2(:,find(n2));
y = unique(data1(:,3));
x =[(n1./sum(n1))' (n2./sum(n2))'];

  set(0,'defaultaxesfontsize',20);
  subplot(2,2,3); bar(y,x,.25,'hist'); % <- percentage cum dist
     ylabel('\bf\fontsize{20} Proportion');
     xlabel('\bf\fontsize{20} Values')
     title('\bf\fontsize{20} Race')
     legend('Sampled Data','Original Data')
grid on

     %% Variable 4
     n1=hist(data1(:,14));
n1 = n1(:,find(n1));
n2=hist(data2(:,14));
n2 = n2(:,find(n2));
y = unique(data1(:,14));
x =[(n1./sum(n1))' (n2./sum(n2))'];

  set(0,'defaultaxesfontsize',20);
```

```
  subplot(2,2,4); bar(y,x,.25,'hist'); % <- percentage cum dist
     ylabel('\bf\fontsize{20} Proportion');
     xlabel('\bf\fontsize{20} Values')
     title('\bf\fontsize{20} Cancer')
     legend('Sampled Data','Original Data')
grid on



% Algorithm to expand data set using count column

% BCdata = importdata('BCdata.txt','\t',1)  % load Data
BCexpand = zeros(2392998,16); %create new matrix for expanded data set
numold = 1;
for i = 1: size(BCdata.data,1)
    num = BCdata.data(i,16);
    row = BCdata.data(i,:);
    mat = repmat(row,num,1);
    BCexpand(numold:numold+(num-1),:)= mat;
    numold = numold+num;
end



% program to compute normed KL distance between two data sets(joint
% probability distributions

function [dnorm,dist] = normKLDiv(P,Q,maxbin)
 clc
 p = multprob(P,maxbin); % distribution of true distribution
 q = multprob(Q,maxbin);% distribution of estimated distribution
 m = size(p,2);
 dist = zeros(1,m);
 for i = 1:m
     p1 = p(:,i);
     q1 = q(:,i);
     pdist = p1(find(p1));
     qdist = q1(find(q1));
     dist(i) = sum(pdist.*log(pdist./qdist));
 end
 logdist = log(dist);
 dnorm = norm(logdist,2); % computes the Euclidean norm of the
distribution vector
% program to compute the probability vector of a multinomial
distribution probability distributions

function p = multprob(x,maxbin)
% p = multprob(x1,x2) computes the probability distribution of the
% multinomial random variable x
% Input: x = n x m matrix of m random variables and n cases of each
%        maxbin: the maximum bins for all variables in distribution
```

```
% Output: p: maxbin x m matrix containing probability distribution of
each
% random variable

m = size(x,2);
p = zeros(maxbin,m); % zeros vector for probability distribution
for i = 1: size(x,2)
   u = unique(x(:,i));
    for j= 1: length(u)
        p(j,i) = length(find(x(:,i) == u(j)))/length(x(:,i));
    end
end
```

Bibliography

Barlow WE, W. E., Ballard-Barbash R, Vacek PM, Titus-Ernstoff L, Carney PA, Tice JA, Buist DSM, Geller BM, Rosenberg R, Yankaskas BC, Kerlikowske K. Prospective (2006). Breast Cancer Risk Prediction model for women undergoing screening mammography. J. N. C. Institute.

Chickering, D. M. (2002). "Learning Equivalence Classes of Bayesian- Network Structures." Journal of Machine Learning Research: 445-498.

Chickering, D. M. (2002). "Learning Equivalence Classes of Bayesian-Network Structures." Journal of Machine Learning 2: 445-498.

Clark Glymour, R. S., Peter Spirtes, Joseph Ramsey. "The TETRAD Project." from http://www.phil.cmu.edu/projects/tetrad/current.html.

David Heckerman, D. G., David M Chickering (1995). "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data." Machine Learning(20): 197-243.

Gail, M. H. (1989). "Projecting Individualized Probabilities of Developing Breast Cancer for White Females Who Are Being Examined Annually." Journal of the National Cancer Institute 81: 1879-1886.

Guoliang LI, T.-Y. L. (2007). "Biomedical Knowledge Discovery with Topological Constrainst Modeling in Bayesian Networks: A Preliminary Report."

Institute, N. C. (2010). "National Cancer Institure Fact Sheet." from http://www.cancer.gov/cancertopics/factsheet/detection/probability-breast-cancer.

Jeffrey A. Tice, M., Steven R. Cummings, MD, Rebecca Smith-Bindman,MD et al (2008). "Using Clinical Factors and Mammograph Breast Density to Estimate Breast Cancer Risk: Development and Validation of a new Predictive Model." Annal of Internal Medicine 5(148): 337-347.

Jung Hun Oh, J. C., Rawan Al lozi, Manushka Vaidya, Yifan Meng, Joseph O Deasy, Jeffrey D Bradley, Issam El Naqa (2011). "A Bayesian network approach for modeling local failure in lung cancer." Physics in Medicine and Biology 56: 1635-1651.

Mozaherul Hoque Abul Hasanat, D. R., Rajeswari Mandava (2010). "Bayesian belief network learning algorithms for modellng contextual relationships in natural imagery: a comparative study " Artficial Intelligence 34: 291-308.

Pearl, J. (2000). Causality: Models Reasoning and Inference, Cambridge University Press.

Peter Spirtes, C. G. a. R. C. (2000). Causation Prediction and Search. Cambridge Massachusetts, The MIT Press.

Ramsey, J. (2010). Bootstrapping the PC and CPC Algorithms to Improve Search Accuracy, Carnegie Mellon University.

Richard J Santeen, N. F. B., Rowan T Chlebowski (2007). "Critical assessment of new risk factors for breast cancer: considerations for development of an improved risk prediction model." Endocrine -Related Cancer 10(1677): 169-187.

Schwarz, G. (1978). "Estimating the Dimension of a Model." Annals of Statistics 6(2): 461-464.

Sun-Mi Lee, P. A. A. (2003). "Bayesian networks for knowledge discovery in large datasets: basics for nurse researchers." Journal of Biomedical Informatics 36: 389-399.